HUMAN FACE DETECTION FROM COLOR IMAGES BASED ON MULTI-SKIN MODELS, RULE-BASED GEOMETRICAL KNOWLEDGE, AND ARTIFICIAL NEURAL NETWORK

SINAN A. NAJI

THESIS SUBMITTED IN FULFILLMENT OF THE

REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

UNIVERSITY OF MALAYA

KUALA LUMPUR

2013

UNIVERSITY MALAYA

ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: SINAN A. NAJI **(I.C/Passport No:** G2228117**)**

Registration/Matric No: WHA040026

Name of Degree: DOCTOR OF PHILOSOPHY

Title of ~~Project Paper/Research Report/Dissertation/~~Thesis ("this Work"):

HUMAN FACE DETECTION FROM COLOR IMAGES BASED ON MULTI-SKIN MODELS, RULE-BASED GEOMETRICAL KNOWLEDGE, AND ARTIFICIAL NEURAL NETWORK

Field of Study: ARTIFICIAL INTELLIGENCE - BIOMETRICS

I do solemnly and sincerely declare that:

(1) I am the sole author/writer of this Work;
(2) This Work is original;
(3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;
(4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
(5) I hereby assign all and every rights in the copyright to this Work to the University of Malaya ("UM"), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;
(6) I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature Date:

Subscribed and solemnly declared before,

Witness's Signature Date:

Name:
Designation:

ABSTRACT

Automatic human face detection is becoming a critical required step in a wide range of applications such as face recognition systems, face tracking, content-based indexing retrieval systems, communications and teleconferencing, and so on. The first important step of such systems is to locate the face (or faces) within the image.

This research produces an efficient state-of-the-art system for the detection of frontal faces from colored images regardless of scale, position, illumination, number of faces and complexity of background. The general architecture of the proposed system consists of three main stages: skin detection, face-center localization, and neural network-based face detector.

In the first stage, we use image segmentation techniques to locate human skin color regions in the input image. First, the source image is converted to HSV color space. Then, multi-skin color clustering models are used to detect skin regions in image(s). A total of 24,328,670 training pixels are used to build our skin-models. These pixels are collected manually from true human skin regions using four public databases. The classification boundaries are transformed into a three-dimensional look-up table to speed up the system. Automatic illumination correction step is used for skin color correction to improve the general face appearance.

In the second stage, a rule-based geometrical knowledge is employed to examine the presence of face by locating the basic facial features. The goal of this step is to remove false alarms caused by objects with the color that is similar to skin color. First, the facial features are extracted from skin-maps. Then, rule-based geometrical knowledge is employed to describe the human face in order to estimate the location of the "face-center".

In the last stage, neural network-based face detector is used to decide whether a given sub-image window contains a face or not. The neural network-based face detector is applied only to

the regions of the image which are marked as candidate face-centers. The classification phase consists of four steps: the cropper, histogram equalizer, texture-analyzer, and ANN-based classifier. The function of the cropper is to crop a sub-image's pyramid from the source image. Histogram equalizer is used to improve the contrast. Texture-analyzer is used to compute texture descriptors. Training neural network is done offline and designed to be general with minimum customization. A total of 40,000 face and non-face images are collected for training the ANN-based classifier.

The implementation of different methodologies in one integrated system, where one method can compensate for the weaknesses of another, depicts reasonably accurate results. The system has been trained, tested and evaluated using five public databases which contain faces of different sizes, ethnicities, lighting conditions, and cluttered backgrounds. Comparison with state-of-the-art methods is presented, indicating that our system shows viable detection performance.

ABSTRAK (MALAY)

Sistem pengesanan muka manusia secara automatik semakin menjadi keperluan yang penting bagi aplikasi yang meluas seperti sistem pengecaman muka, pengesanan wajah, sistem pencapaian indeks bagi informasi, komunikasi dan telekomunikasi dan sebagainya. Langkah pertama bagi sistem seperti ini adalah untuk menentukan lokasi bagi satu atau pelbagai muka di dalam sesuatu imej.

Tesis ini memperkenalkan satu sistem yang berkesan bagi pengesanan muka yang berwarna dari sudut hadapan tidak kira dari kiraan skala, posisi, iluminasi, bilangan muka dan komplikasi dari pemandangan belakang. Rangka kerja am bagi sistem ini terbahagi kepada tiga peringkat utama: pengesanan kulit, mengenalpasti muka dari sudut tengah dan rangkaian neural.

Pada fasa pertama, teknik segmentasi imej digunakan untuk mengesan kumpulan kulit dari imej yang diberikan. Yang pertama, imej asal harus di tukarkan kepada skala berwarna iaitu HSV. Kemudian, kluster kepada model pelbagai warna kulit digunakan untuk mengesan kumpulan kulit pada imej tersebut. Sebanyak 24,328,670 sampel kumpulan kulit telah digunakan untuk membina model ini. Sampel-sampel ini di ambil dari sampel kulit manusia yang diambil dari empat kumpulan yang berbeza. Klasifikasi sampel kulit telah ditukar kepada tiga dimensi di dalam bentuk rajah umtuk tujuan mempercepatkan masa. Iluminasi secara automatik digunakan untuk pembetulan warna kulit bagi memperbaiki keseluruhan bentuk muka.

Dalam fasa kedua pula, untuk mengkaji kedudukan sebenar muka dengan mengandaikan lokasi bentuk muka, pengetahuan bagi peraturan geometri digunakan. Tujuannya adalah untuk menghapuskan perkara kesilapan yang ditimbulkan oleh objek yang mempunyai warna yang seakan sama dengan yang asal. Pertama, muka asal di keluarkan dari peta-kulit. Kemudian, pengetahuan bagi peraturan geometri digunakan untuk menerangkan muka manusia untuk mengandaikan lokasi sebenar pusat-muka.

Pada peringkat akhir, rangkaian neural bagi pengesanan muka digunakan untuk menentukan samada rangka muka mempunyai muka atau pun sebaliknya. Klasifikasi ANN digunakan keatas kumpulan imej yang telah ditentukan sebagai calon bagi pertegahan muka. Klasifikasi ini terbahagi kepada empat langkah: pemotong, histogram ’equalizer’, analisa tekstur, dan klasifikasi ANN. Fungsi pemotong adalah untuk memotong imej dari imej asal. Histogram ’equalizer’ digunakan untuk memperbaiki kontrast. Analisa tekstur digunakan untuk mengira tekstur. Latihan rangkaian neural dilakukan secara offline dan di rekabentuk secara umum dengan ‘customization’ paling minima. Sebanyak 40,000 muka dan bukan muka di kumpul untuk latihan bagi klasifikasi ANN.

Implimentasi bagi kaedah yang berbeza dalam satu sistem dapat memberi keputusan yang baik. Sistem ini telah diuji dan dikaji menggunakan lima kumpulan data yang mengandungi muka dari saiz berbeza, etnik, cahaya bagi pelbagai kluster dan latar belakang. Perbandingan dengan kaedah lain yang terkini telah juga dibuat dimana sistem ini memberikan pencapaian yang lebih baik.

ACKNOWLEDGEMENT

I would like to express a great thankfulness to my supervisor, Prof. Dr. Roziati Zaiuddin as well as my co-supervisor, Associate Prof. Dr. Sameem A. Karem for their support, guidance, suggestions and encouragement over the past years of this research. They give me the opportunity to carry out my research with little obstacles. The comments from both supervisors had a significant impact on this thesis. Their unswerving devotion to integrating multi approaches has helped me appreciate how valuable these approaches are. Their help and support in several ways have always been in my mind.

My great appreciation to Dr. Hamid A. Jalab for his valuable discussions, research perspective, suggestions and for his valuable time he spent with me. My sincere gratitude to Prof. Dr. Jubair Al-Jaafer for his scientific help during the research. I would also like to thank Dr. Chee Seng Chan for his comments and suggestions to improve this research.

Valuable comments received from some of the editorial board experts in peer-reviewed journals such as EURASIP Journal on Image and Video Processing, Advances in Complex Systems, Digital Signal processing, and IET image processing have given me insight into the need for robustness in image processing algorithms and their valuable comments to achieve high quality research.

I would like to thank the Faculty of Computer Science and Information Technology, University of Malaya for providing me a great academic environment.

And finally, closer to home, thanks to my wife Iftekhar Fadhil for her constant love and support throughout the years required to complete my PhD. My deepest gratitude to my parents for their encouragement and support during this research.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

Adaboost	Adaptive boosting
AI	Artificial Intelligence
ANN	Artificial Neural Network
AVG	Average
CIE	Commission Internationale de L'Eclairage (the International Commission on Illumination) Color Space
CMU	Carnegie Mellon University (Face database)
CMY	Cyan, Magenta, and Yellow Color Space
DIP	Digital Image Processing
DT	Decision Trees
DWT	discrete wavelet transform
EM	Expectation-Maximization Algorithm
FN	False Negative
FP	False Positive
GMM	Gaussian mixture model.
HMM	Hidden Markov Model
HSV	Hue, saturation, and Value Color Space
JAFFE	The Japanese Female Facial Expression Database
LBP	Local Binary Patterns
LDA	Linear Discriminating Analysis
LGP	Local Gradient Patterns
M2VTS	Multi Modal Verification for Teleservices and Security applications
MIT	Massachusetts Institute of Technology (Face database)
ML	Machine Learning
NN	Neural Network
PCA	Principal Component Analysis

PR	Pattern Recognition
RGB	Red, Green , And Blue Color Space
SD	Standard Deviation
SGM	Single Gaussian Model
SL	Sign Language
SNoW	Sparse Network of Winnows `
SOM	Self-Organized Maps
SVM	Support Vector Machines
TN	True Negative
TP	True Positive
TSL	Tint, saturation and lightness colour space
YCbCr	Luma (Y) and two chrominance (CbCr) components Color Space
YUV	Luma (Y) and two chrominance (UV) components Color Space

CHAPTER ONE

INTRODUCTION

1.1 Research Inspiration and Background

Vision is the most advanced of our senses, so it is not surprising that images play the single most important role in human perception (Gonzalez, Woods, & Eddins, 2007). It is estimated that 90 to 95% of the information received by a human is visual (Russ, 2007). We receive visual information from the world around us via our vision system and are able to recognize objects with practically no effort. Humans constantly detect/recognize objects such as people, buildings, etc. Yet it remains a mystery how the human brain detects and recognizes objects (Zhang & Zelinsky, 2004). The vision system is the most used of human senses for identifying individuals by their faces (or face images) when we see them.

In modern life, identification and authentication are frequently required as initial procedures for various daily processes. In classical up to date computer systems, the procedure for identifying users (authentication) is based on something that one knows (such as user name and password), or something that one carries (such as a magnetic card, key, or chip card). With the ubiquitous current technology available, these methods, however, are no more secure for identity verification. The passwords can be easily disclosed, broke into, or forgotten. Cards may be stolen or lost. To achieve the verification (or identification) process in better, more trustworthy ways, and to reduce the fraudulent claims of individuals, we should use something that actually discriminates the given person. Biometrics offer efficient ways for verifying the identity of humans by their characteristics based on the principle of measurable physiological and/or behavioral characteristics of individuals (Vaclav & Zdenek, 2001). Characteristics such as fingerprints, face, hand geometry, iris, voice, and signature dynamics are unique to every individual and can be used for human biometric verification and/or identification methods.

Unfortunately, most biometric methods have yet to gain acceptance by the general population for two reasons:

1) People do not like to give samples of their own characteristics every time; and
2) People do not like to use things that are used by others. For instant, the same biometric reader or scanner is used to take samples from a group of individuals.

Up to now, face recognition systems can be viewed as the most successful application of biometric methods which have gained significant attention (Li & Xu, 2009). Almost all recognition systems assume that the input human face has been correctly detected and cropped during the preprocessing stage. In general, the problem of face detection is all about face recognition. This is a fact that seems quite bizarre to new researchers in this area. However, before face recognition is possible, the system must be able to reliably find the face and its landmarks in the input image. Compared with other biometric methods, face recognition is more convenient, non-intrusive, and more acceptable for users. With such a system, a camera (maybe hidden-camera) can be used to obtain the face image of an individual even without prior knowledge. These cameras are usually used in public places such as airports, banks, etc. Therefore, it is the most natural mean of biometric identification. Recently, the cost of computer peripherals such as digital cameras has become affordable to most users. The computer now, has the ability which many of us have taken for granted, that is, the ability to see and analyze.

The digital camera opens the field of "Computer Vision" which involves the study and application of methods for computer systems to get information from image content to perform a specific task with the ultimate aim of using machines to emulate human vision (Gonzalez & Woods, 2002). The research into computer vision dates back to the 1960's when most of low-level image processing were proposed (Gonzalez & Woods, 2002). Although excellent outcomes have been obtained in other artificial intelligent fields (e.g. natural language processing, expert systems, and game playing), computer vision still seems to have lagged behind in many aspects. In the broadest possible sense, image (or picture) is a way of recording and presenting information "visually". Pictures are important to us because they are an effective

medium for saving information and also can be used, concomitantly or later, for communication. There is thus a scientific basis for a well-known saying that "a picture is worth a thousand words" (Efford, 2000). In the early times, humans could do this only in the form of drawings. The invention of photography triggered the first revolution in the use of images. The Daguerreotype process invented by the French painter, Louis Daguerre in 1839 became the first commercially utilized photographic process (Jahne, 2004).

Digital cameras constitute the second revolution in the use of images. Image processing is a rapidly growing area of computer science. The current technological advances in imaging equipment, high speed processors, and high capacity of storage units have raised the growth of digital imaging. Nowadays, most fields that are based on traditional analog imaging such as medical images, film productions, etc. have shifted to digital systems. In general, we are generating a huge amount of images, with various forms and complexity, more than could ever be examined manually. Becoming digital may have been inevitable, but a major accelerating factor in this change has surely been the internet or, more specifically, the World Wide Web which provides the medium through which millions of images are moved daily at all points of the globe (Efford, 2000). The amount of digital video available for many applications has undergone explosive growth in recent years more than could ever be organized and indexed manually. Through video indexing we are able to tell whether or not a specific object of interest (e.g. specific actor) is present in a video sequence shot or not (Albiol, Torres, Bouman, & Delp, 2000; Clippingdale & Fujii, 2011; Doudpota & Guha, 2012). Due to the huge amount of image collections and videos, there is a need for automatic detecting and/or recognizing faces in images such that users can find them quickly. Unfortunately, to date the usability of the large image collections is limited and there is a clear shortage of efficient image retrieval methods. Currently, text-based image captions (i.e. surrounding words) or low-level features such as color, texture and shape are used to find a specific image in such a collection (Lei, Peng, & Yang, 2012).

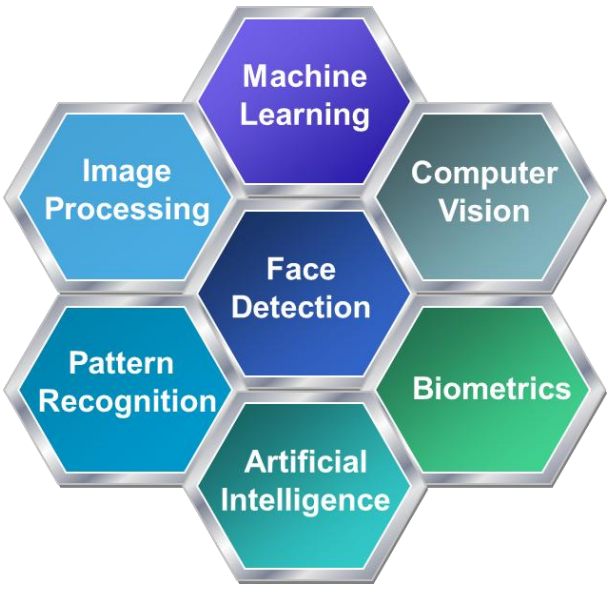


Figure 1.1: Relation between Face Detection and various other fields.

One final goal of image analysis is to automatically detect/recognize real objects or scenes. For many applications, simply knowing the presence or absence of an object is useful. One of the major problems in the design of modern face processing systems (e.g. face recognition) is automatic face detection from images.

Face detection is becoming an active research area spanning several disciplines such as image processing, pattern recognition, machine learning, computer vision, artificial intelligence, and biometrics (see Figure 1.1).

By taking one step ahead into the problem, it is important for any face processing system to locate faces automatically, quickly and accurately. Therefore, automatic face detection is a key problem and a necessary first step in many applications such as face recognition systems, content-based indexing retrieval systems, robotics, human computer interface, expression estimation, communications (e.g. video phones) and teleconferencing, facial expression recognition and gender recognition.

1.2 Automatic Face Detection

We can define face detection from images as follows:

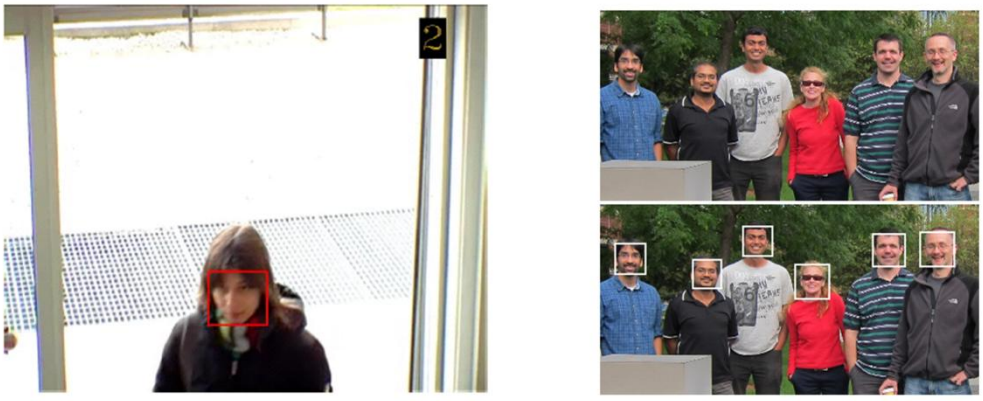
> *Given an arbitrary image, the aim of face detection is to develop computer systems that can mimic human's ability to find human face (or faces) in an arbitrary image and, if present, return the location and extent of each face* (Yang , Kriegman, & Ahuja, 2002).

The human face detection problem can be divided into two categories (see Figure 1.2):

1) Face detection in static images, and

2) Real-time face detection (or sequence of video images).

The first category, which forms the theme of this thesis, is more complicated and harder because of the diversity of image types and sources (see Section 1.3). In real-time systems the face detection problem is simpler than static images because it provides more information such as a series of image frames from video-camera, as shown in Figure 1.2(a). By comparing the sequence of images one can extract the moving objects such as human targets since most of the surrounding background is fixed. Furthermore, in many real-time face detection systems, the developers usually control the environment in which the detection system will take place. Predefined assumptions, using other range of sensors, and restrictions that are used by developers simplify the detection problem.



(a) Real-time face detection

(b) Face detection in still images

Figure 1.2: Automatic human face detection; (a) face detection in real-time applications; (b) face detection in static images.

Numerous methods to detect faces in still images are presented in the literature. These methods can be classified according to the input into three categories:

- 2D gray scale images (Guo & Wu, 2010; Nefian, 1999; Rowley, Baluja, & Kanade, 1998; Turk & Pentland, 1991a; Viola & Jones, 2004).
- 2D colored images (Hiremath & Danti, 2006; Jin , Lou, Yang, & Sun, 2007; Moallem, Mousavi, & Monadjemi, 2011; Shih, Cheng, Chuang, & Wang, 2008).
- 3D depth images (Colombo, C., & R., 2006; Nair & Cavallaro, 2009; Niese, Al-Hamadi, & Michaelis, 2007; Schneiderman & Kanade, 2000).

Most of the images we really want to process are essentially two-dimensional colored images (Russ, 2007), see Figure 1.2(b), which are considered in this thesis to be our research focus due to their immense ubiquitous existence, yet requiring solutions to many issues and in need of effective processing.

From the point of methodology, numerous approaches had been proposed (see Chapter 2). Each method has its strengths and weaknesses. The accuracy of face detection systems should improve with time, but it has not been very satisfying so far. Figure 1.3 shows the output from a state-of-the-art face detector named: FaceSDK software with default setting, executed as an experiment on 9th Dec. 2011 at Faculty of Computer Science and Information Technology, University of Malaya. In Figure 1.3(a), the face detector has correctly detected a face. The second example is shown in Figure 1.3(b). This figure shows the opposite situation. Here, the face detector has missed many faces, while it has incorrectly classified a patch of ground to be a face.

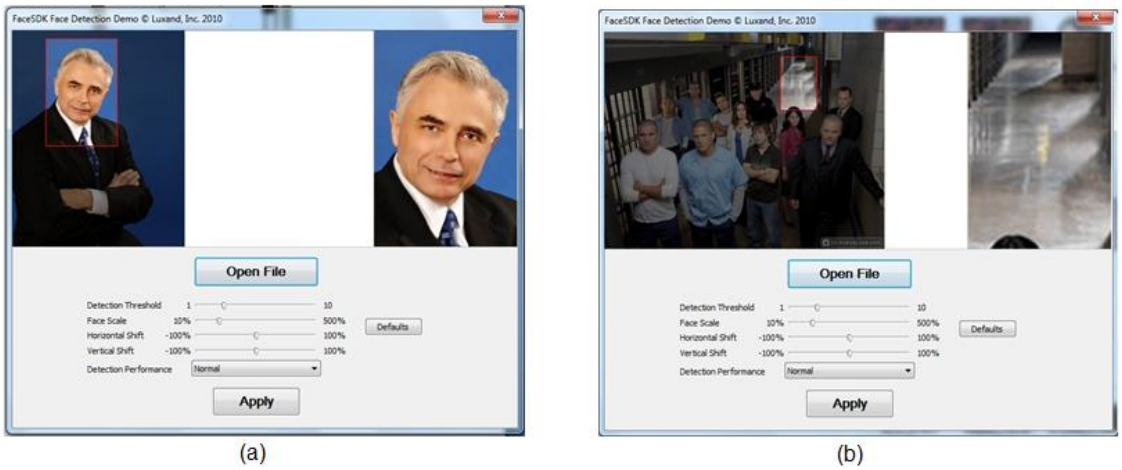


Figure 1.3: Face detection example, FaceSDK Software using default setting executed on 9th Dec. 2011. (a) Positive detection; (b) False detection.

However, face detection problem is a popular active research area in both academic and commercial institutes. Reasons for this trend:

- There is a real-world need for such face processing systems which are needed in civil governmental foundations, the military, commercial applications, etc. Table 1.1 shows some examples of face detection in application to authentication/identification systems.
- After many years of research and efforts, the supporting technologies have progressed to the point where the use of this technology is now viable.

The research interest in face detection problem is shown by many specific face processing conferences such as Automatic Face & Gesture Recognition (AFGR), The International Joint Conference on Biometrics (IJCB), Audio and Video-Based Biometric Person Authentication (AVBPA), and systematic empirical supporting materials including many face databases available on-line for researchers such as the FERET, XM2VTS , CVL, JAFFE, LFW, and several others ("XM2VTS Face Database," 2102) ("FERET Face Database," 2012) ("CVL Face Database," 2012) ("JAFFE Face Database," 2012) ("LFW Face Database," 2012).

Table 1.1: Examples of face detection in application to authentication/identification systems.

Access Gates	Face image is used in place of (or with) the traditional methods to gain access into secure restricted areas such as buildings, labs, control rooms, etc.
Police	Face image can be regarded as a unique physiological measure in investigations and crime cases.
Immigration	Reduce unlawful entry for individuals in Airport check-ins and other check-in points at country borders.
ATM Access	Enhance the trustworthiness of financial transactions done through ATM machines
Equipment Usage	To restrict the usage of equipment such as laboratory equipment, computers, and control devices.
E-commerce	To increase the reliability of financial transactions done through e-commerce
Suspects Verifications	To speed up authentication and identity verification in public places without participant's cooperation or even prior knowledge.
Government Agencies	To reduce the fraudulent claims of individuals such as an impostor pretending to be a client.
Content-Based Image Retrieval (CBIR)	Instead of using text-based captions, a face image of a specific individual is used to find image(s) in large database(s) based on similarity of his face image. Also, to find human targets in new images and create textual captions based on image's contents.
Web Search Engines	Same as CBIR to be used by current Web search engines such as Google image search, Lycos and AltaVista photo finder.
Video Databases Indexing Applications	To tell whether or not a specific individual is present in a video sequence shot or not.

The goal of this research is to solve the face detection problem efficiently and accurately by solving many sub-problems implied in the main problem using a hybrid system that encompasses different methods within its structure. The system should cope with the main varying random factors such as different lighting conditions, ethnicities, and complex backgrounds. It must be able to handle dark skin tones and adjust skin darkness to improve the performance of the face detector.

1.3 Problem Statement

Although human faces have the same facial constitutes and structure that can be realized and described easily, their appearance in 2D images shows a high degree of variability that does not permit rigid model-based description. For images of realistic complexity, the human face is a dynamic object in its appearance (not a rigid object), which makes face detection a difficult problem in computer vision. The shortage of the current systems is clearly noticeable when compared to the detection capability of humans. We can detect faces from images almost instantly and our own recognition ability is superior to that of the computer systems. The scaling differences and different complex backgrounds do not affect our ability to detect faces.

The main factors that make face detection by a computer system a challenging task can be attributed to the following (Moallem et al., 2011; Yang et al., 2002):

- **Number of faces in the source image:** single or multiple faces may appear in the image.
- **Scale:** faces might have unknown size due to different distance from camera.
- **Location**: faces can appear anywhere in the image.
- **Rotation**: in practice, we found that most natural scene faces are rotated about ±10° (i.e. inclination).
- **Pose**: The appearance of the face in images differs due to the face pose (frontal, profile, etc.).
- **Presence or absence of facial features**: mustaches, beards, glasses can dramatically change the appearance of an individual's face.
- **Shape variation**: Variations in the shape of an individual's face. This type of variation includes facial constitutions, whether it has a long nose or short, wide eyes or small, the eyebrow and eye are close to each other or isolated, etc.
- **Facial expression**: Facial expressions are appeared in the state of Happiness, Anger, Sadness, Surprise, Fear, and Disgust. These are responses that appear on the human face due to the facial muscles contraction. Variations due to facial expressions directly affect the appearance of faces in images.

- **Occlusion**: some parts of faces could be occluded by other objects.
- **Lighting conditions**: Varying illumination, non-uniform lighting, and shadows may cause various kinds of effects on the face. This is due to the non-plane shape of the facial features. Changes in the light source in particular can radically change a face's appearance.
- **Complex background**: the diversity of the background is virtually unlimited such as clothes, furniture, buildings, etc.
- **Non-face definition**: from the point of classification, we have a two-class classification problem that is, face and non-face image, which are not equally complex. It is easy to get face images but it is hard to get representative samples of "non-face" images.
- **Different ethnic groups**: faces of people from different racial groups usually appear differently.
- **Losing of information**: When we look at an object from different viewpoints we perceive different images. In processing 2D digital images, information is already being lost when transforming the 3D world to 2D image. It is difficult to reconstruct the actual 3D representation from an arbitrary image.
- **Image reproduction**: Scanned images, internet images, newspapers images, etc. are usually uncontrollable and have virtually unlimited sort of reproduction and montage processes.
- **Camera characteristics**: different color cameras do not necessarily produce the same color appearances for the same scene.

Examples of such variations are shown in Figure 1.4. Apparently these variations complicate face detection and the larger the variations are, the more difficult the problem is. Since face detection in a general setting is a very difficult task, application systems typically restrict one or many aspects, including the environment in which the detection system will take place. These systems usually put preconditions and assumptions such as uniform lighting, single face, frontal face, uniform background, and dark clothes.

Figure 1.4: Variations in face appearance complicate face detection.

Object detection is difficult in general, because complex images contain huge amounts of information at different levels. In digital image processing systems, images are arrays of numbers created from the physical scene picture (Salah, Bicego, Akarun, Grosso, & Tistarelli,

2007). With the current technology, the digital image may contain millions of pixels; each pixel may carry important information. To detect objects with minimum error, we would have to build an efficient classifier that can cope with variations in the object's appearance (i.e. variations in pixels intensities).

In this research, we propose to use a neural network-based classifier for final classification with other integrated methodologies, but for the moment let us consider one of the fastest classification methods, that is, the Lookup-Table. An ideal Lookup-Table based classifier with an entry for every possible input would show relatively accurate classification with minimum classification errors. The table is constructed offline and indexed by the combination vector. Each row contains information indicating its classification output that is Face or Non-Face class. For example, consider a thumbnail 20×20 pixels face pattern (i.e. gray scale) shown in Figure 1.5, then we have $256^{400} \cong 10^{963}$ possible entries in this Lookup-Table that is required for classification of 20×20 pixels region, which is enormously high dimension. Table 1.2 shows such a Lookup-Table representation that would be indexed for all the possible combinations. It is clear that such a table is impracticable. Speed and memory limitations push us to seek other solutions such as a neural network-based classifier or other methods.

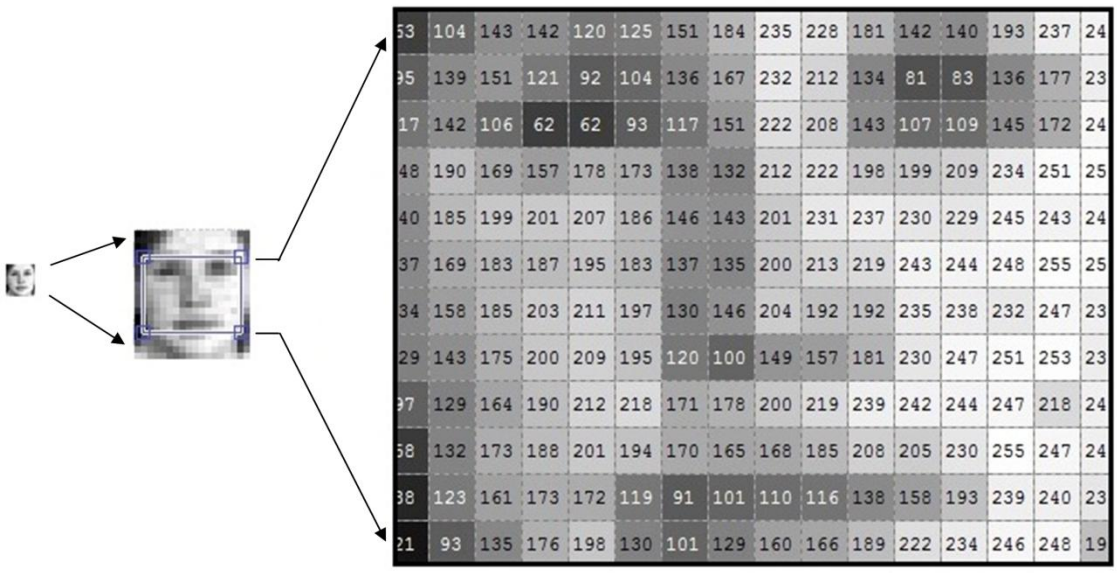


Figure 1.5: Thumbnail face pattern of size 20×20 pixels.

Table 1.2: An ideal Lookup-Table for face detection used for a face pattern 20 × 20 pixels shown in Figure 1.5.

	1 $f(1,1)$	2 $f(1,2)$	3 $f(1,3)$	……..	399 $f(20,19)$	400 $f(20,20)$	Classification
1	0	0	0	……	0	0	Non-face
2	0	0	0	……	0	1	Non-face
3	0	0	0	……	0	2	Non-face
.	……	……	……	……	……	……	……
.	……	……	……	……	……	……	……
.	163	164	165	……	146	150	Face
	163	164	165	……	146	151	Face
.	……	……	……	……	……	……	……
10^{963}	255	255	255	……	255	255	Non-face

However, we can get practical results in object detection by reducing the problem's generality. We can restrict the domain of the problem to a constrained environment by imposing preconditions and assumptions about the types of objects, size, lighting, background, etc. Therefore, many researchers focus on obtaining practical results to limited types of images instead of designing a system that can work for all types of images. For example, some techniques assume the availability of passport-like images.

In developing an appearance-based object detector that uses machine learning, many subproblems arise. The problems targeted by this research are summarized as follows:

- **The huge search space:** from the point of the traditional appearance-based face detectors, the classification process is best known as the sliding-window technique. A sliding window is applied at every pixel location in the source image and at multi-scale pyramids. Although many previous works proposed to speed up the implementation by introducing an incremented step during moving of the sliding window, the search space is still high.

- **Illumination variations**: is one of the significant factors affecting the appearance of the face in image(s). The existing illumination enhancement methods are inadequate.
- **Quality of training data**: the traditional ways for data preparation imply high variations in the training face images which degrade the quality of data.
- **Discrimination ability:** feature vector based on pixels intensities alone is inadequate to construct a powerful classifier. If we do not augment the input vector with adequate features, even the most sophisticated classifiers may fail to accomplish the classification task.
- **Reliability of face detector:** the natural non-face patterns which are similar to face patterns confuse the classifiers. Since we use small patterns for the training and classification, the existence of such patterns in the source image increases the false detections. Figure 1.6 shows an example of a natural pattern that looks like a face when considered in separation (i.e. false detection) adopted from Sung and Poggio (1998).
- **Accuracy**: most skin color modeling and detection methods show high False Negative (FN) and/or False Positive (FP) errors when dealing with images captured under unconstrained imaging condition (Kakumanu, Makrogiannis, & Bourbakis, 2007; Tan, Chan, Yogarajah, & Condell, 2012).
- **Speed:** The high computational cost of many skin detection methods and appearance-based object detectors.
- **Evaluation:** An important characteristic underlying the design of image segmentation methodologies is the considerable level of testing and evaluation that normally is required before arriving at the final acceptable solution. Up to date, although there is an enormous amount of research dedicated to image segmentation algorithms, there is a limitation about how to evaluate different segmentation methodologies (Gonzalez et al., 2007; Russ, 2007). What is sought is a formulating guideline for a specific purpose that is detecting human skin using color feature and based on specific standard set of test images.

Figure 1.6: Natural non-face pattern that looks like a face pattern (i.e. false detection) adopted from Sung and Poggio (1998).

1.4 Research Aim and Objectives

The ultimate aim of this work is to develop a face detection system that is capable to locate frontal human face (or faces) with $\pm 10^0$ rotation from color images efficiently and accurately. The system should overcome the sensitivity to the variation in face size, location, ethnicity, lighting conditions, and complex background. Based on the findings of the literature review (Chapter 2), the decision was made on which approaches that need improvements in terms of accuracy and/or computational cost. This research is motivated by different goals with specific objectives as follows:

1) To develop a new skin color modeling and detection method for detecting human targets in complex images.
2) To propose a new methodology for testing and evaluation of image segmentation techniques.
3) To devise a new method for automatic illumination enhancement in application to face detection.
4) To build a rule-based geometrical knowledge for face-center localization.
5) To develop an efficient appearance-based face detector based on machine learning techniques.
6) To test and evaluate each stage of the proposed system under different conditions.

1.5 Research Questions

Several research questions have been formulated to serve as a guideline to conduct this research and to achieve the research objectives. The ten basic questions are:

Q1. How can the challenges and difficulties, which are mentioned before, be solved more successfully? In other words, what kind of a novel classifier is proposed to solve these difficulties?

Q2. How effective and useful is the skin detection approach to face detection?

Q3. What is the suitable color space for human skin-color segmentation?

Q4. How can we generalize the proposed skin segmentation approach so that we can apply it to other applications?

Q5. An important characteristic underlying the design of image segmentation methodologies is the considerable level of testing and evaluation. With the limitations on how to measure segmentation accuracy and error rates, how can we test and evaluate different skin segmentation approaches and determine their performance?

Q6. How can the proposed skin color model suppress the FN/FP errors caused by many random factors?

Q7. What novel method is proposed to carry out automatic illumination enhancement?

Q8. How can the rule-based geometrical knowledge speed up the system?

Q9. What kind of new face model that can be used to improve the detection rate of the classifiers?

Q10. How can the appearance-based face detector be superior to the existing ones in terms of detection rate and speed?

1.6 Scope of Work

This research comprises a number of stages:

- **Research investigation:** Based on the findings of the literature review, the limitations of the current face detectors are identified. Information is gathered from publications including journals, conference papers, and thesis both locally and abroad. Then, decision

was made on which approaches need improvements in terms of accuracy and computational cost.

- **Methodology:** The design and implementation of the proposed system integrate three main different methods within its structure, these are: skin detection stage based on skin color feature, rule-based face localization, and neural network-based face detector (Section 1.7).

- **Data collection:** The data collection at the skin detection stage is conducted using three public face databases, these are: FEI dataset, CVL dataset, and LFW dataset. Due to some limitation of the above-mentioned databases, we developed our own database in this research to serve the purpose. Skin and non-skin samples are prepared manually. The dataset is composed of more than 20,000,000 pixels. The data collection for the ANN-based classifier is conducted using the above-mentioned datasets plus JAFFE face database. All training faces are prepared with semi-automatic method. The dataset is composed of 40,000 images. There are 20,000 positive samples of face patterns and the rest are non-face.

- **Conducting experiments:** Qualitative and quantitative results on the above-mentioned datasets have been obtained. Generally, when proposing a system with composite steps, it is interesting to evaluate each step separately. The performance evaluation of each method is conducted under different conditions, such as complex background, illumination variation, ethnicity and a comparison with the state-of-the-art methods in terms of time and accuracy.

- **Documentation:** Some of this research ideas and findings are reported in publications (see Appendix-A). The whole research is documented in this thesis.

1.7 Research Methodology

In practice the problems associated with face detection, especially the processing of complex images can rarely be successfully solved through the application of just one methodology or one classifier (Frisch, Vrschaeb, & Olanoc, 2007; Sonka, Hlavac, & Boyle, 2008). This makes face detection a challenging issue in computer vision. The implementation of different methodologies in one integrated system, where one method can compensate for the weaknesses of another, will improve the general performance of the system to achieve the desired goals. In this research, the principal problem is divided into several more manageable sub-problems that, when solved using different approaches, can resolve the main problem. Accordingly, the system is designed in such a way that it is based on a set of cascade classifiers where each classifier rejects non-face regions (or pixels) based on different features such as color, intensity, textures, etc. The main advantages of using a set of cascade classifiers with different features are:

- To restrict the search space of the subsequent complex classifiers and consequently speed up the system. Fast, simplified, and reliable classifiers are used first to reject the majority of the image's pixels prior to calling the more complex classifiers.
- To increase the reliability of the system. Backgrounds usually contain many natural non-face objects/patterns in the real world which look like face pattern. Excluding the background is an important step to avoid the majority of false detections.

The general architecture of the proposed system consists of three main steps as shown in Figure 1.7, these are:

- Skin detection (or image segmentation based on color feature) and illumination enhancement.
- Face-center localization.
- Neural network-based face detector (or classifier).

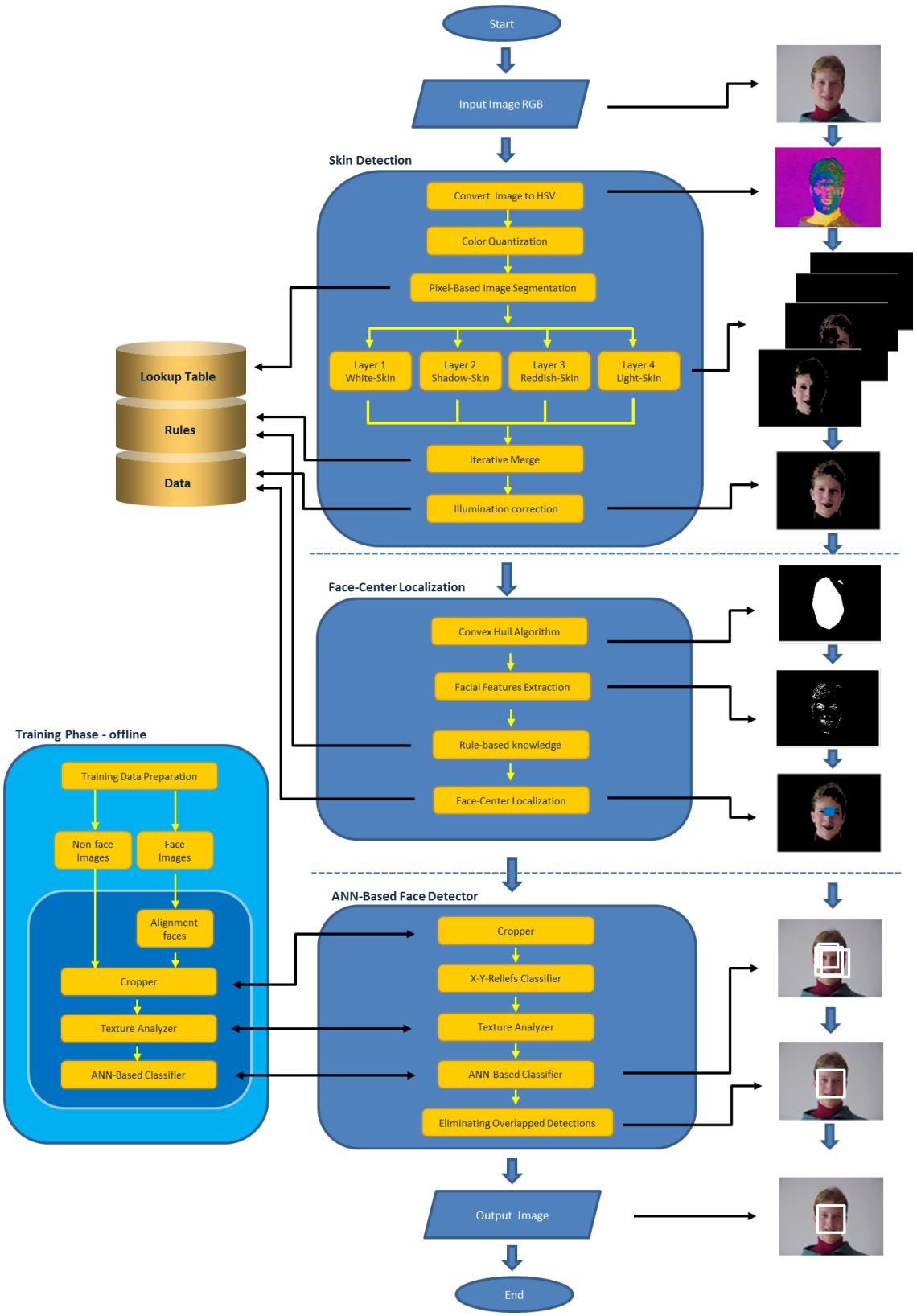


Figure 1.7: The general system architecture.

A brief description of these steps is presented in this section as follows:

- **Skin Detection:** with the ultimate goal of detecting human faces from complex images automatically, it is suggested to use human skin detection techniques as the first step in the proposed system. It is basically an image segmentation problem as the input image is to be segmented into two parts: one containing human skin regions and the other representing non-skin regions that is background. Our pixel-based image segmentation approach is based on color feature (i.e. human skin color) and it is considered one of the fastest classifiers where each pixel is classified as either skin or non-skin without calculations. Accordingly, the source image is segmented into "regions of interest". Each region is considered a candidate face region that would be passed to the subsequent steps of the system which comprise more complex classifiers. As shown in Figure 1.7, the input image is converted to HSV color space and then color quantization is used to reduce the number of colors. Then, pixel-based image segmentation is used to detect skin regions in the input image(s) and produce four layers of binary images. Our pixel-based skin detector uses a Lockup Table named SD-LUT for classification purpose to speed the system. Then, the iterative merge combined with automatic illumination correction are used to enhance the illumination of skin regions to improve the general face appearance.

- **Face–Center Localization:** in this stage, a rule-based geometrical knowledge is employed to examine the presence of face by locating the basic facial features. The goal of this step is to estimate the location of the "face-center". First, the facial features are extracted from the image. Then, rule-based geometrical knowledge is employed to describe the human face and to estimate the location of the "face-center". When the face-center is located, all the other skin regions (or false alarms caused by objects with the color that is similar to skin color) will be removed.

- **Neural Network-Based Face Detector:** is used to make the final arbitration and decide whether a given sub-image window contains a face or not. The classification phase consists

of four steps: the cropper, histogram equalizer, texture-analyzer, and ANN-based classifier. The function of the cropper is to crop a sub-images pyramid from the source image. Histogram equalizer is used to improve contrast. Texture-analyzer is used to extract texture descriptors. Training neural network is done offline and designed to be general with minimum customization. When a "face" is detected in the input image, the system draws an appropriate bounding box at the corresponding face.

Each step of the system is presented in detail throughout this thesis in a separate chapter.

1.8 Research Contributions

There are four main contributions of this research. The important ideas had been presented throughout this thesis. Although, we apply some ideas for face detection problem, each idea by itself is general and can be used for a variety of object detection and recognition tasks.

The first main contribution of this thesis is a new reliable skin color pixel modelling and segmentation approach published in our research paper *(Naji, Zainuddin, & Jalab, 2012).* The proposed models can overcome sensitivity to variations in lighting conditions, ethnicity, and complex backgrounds. To the best of our knowledge, this is the first attempt that employs multi-skin color clustering models to detect human skin in complex images. An important challenge for any skin segmentation approach is to accommodate different ethnic groups (e.g. European, African, Asian, etc.) combined with varying imaging conditions. A key contribution of our work is the analysis, interpretation, and classification of the collected data into different clusters based on skin color appearance. Our approach is based on building four skin color clustering models, namely: white-skin model, blackish-skin model, reddish-skin model, and light-skin model. The main advantages of this method can be summarized as follows. ***1)*** This method overcomes the limitations of the traditional uni-skin model. When classification boundaries contain information about the skin color of various races and illumination, the probability of missing faces would be reduced. In other words, pixels which are indistinguishable in one model may be fully distinguishable in the other model. ***2)*** To exploit more information about skin regions and

the relationship between regions. This is an important issue in image segmentation. ***3)*** This method of image segmentation takes into consideration the other steps of the system. ***4)*** This method is done without calculations based on look-up table that makes it applicable in real-time systems where the speed is a critical factor.

The second contribution of this thesis is an efficient approach for testing and evaluating pixel-based image segmentation methods which come due to unavailability of such methods (Gonzalez & Woods, 2002; Russ, 2007; Sonka *et al.*, 2008). Testing and evaluation of segmentation techniques give us tools to measure and compare the characteristics of segmentation methodologies, and thus determine their performance. Numerous methods for image segmentation were shown in the literature. It is important to be able to evaluate and compare these methods. In general, evaluation reduces the time and cost required to arrive at practical systems. Our method is based on proposing a standard set of test images, general guidelines for evaluating the performance of image segmentation, and step-by-step evaluation procedure (see Section 4.10).

The third main contribution of this research is a novel illumination normalization method for the pre-processing of face detection under non-uniform lighting conditions published in our research paper *(Naji, Zainuddin, & Al-Jaafar, 2010).* The novel approach could compensate all the illumination effects in face image, like the attached shadows and low lighting. Firstly, it estimates the illumination of part of the face image using statistical properties. Secondly, it adjusts the illumination of the other parts of the face image based on that estimation. The main advantages of this method are. ***1)*** It is based on local image enhancement technique rather than the whole image. ***2)*** It is done prior to the classification phase (i.e. final classifier operation) which makes the system faster.

The fourth main contribution of this thesis is developing an efficient ANN-based face detector (or classifier) published in our research paper (Naji, Zainuddin, Jallb, Zaid, & Eldouber, 2011). It is well known that even with the choice of a particular machine learning technique, the

problem of face detection implies a number of sub-problems that need suitable solutions in order to achieve acceptable performance. The new solutions imply the following. ***1)*** A novel face model has been used in this research, i.e. partial face pattern. To the best of our knowledge, this is the first attempt that employs a partial face model for classification. ***2)*** A new semi-automatic method is proposed to prepare training faces instead of manual preparation. ***3)*** To augment the feature vector of the classifier, a set of texture descriptors are proposed for six regions of the face model. These descriptors are statistical properties and several pair-wise ordinal contrast relationships across facial regions which are relatively insensitive to the illumination variations. This will enhance the reliability of the system.

As part of the development of the system described in this thesis, we collected a large quantum of data such as skin and non-skin samples, face and non-face images for training phase, and face database images that can be used for future work by other researchers.

1.9 Thesis Outline

The remaining chapters of this thesis are organized as follows:

Chapter 2 describes related works in the face detection. The main characteristics of different approaches are also presented.

Chapter 3 presents the background about human skin color modeling and detection methods. It gives an idea about dealing with color spaces and the main advantages and disadvantages of different techniques that use different color spaces.

Chapter 4 describes the methodology of the first stage of our system that concerns image segmentation. The chapter presents a reliable color pixel-clustering model for skin segmentation under unconstrained scene conditions. This chapter also presents our proposed testing and evaluation methodology.

Chapter 5 presents our methodology for automatic illumination correction to the face image appearance for images that are captured in non-uniform lighting conditions. The approach is based on local enhancement of skin color tone rather than the entire image.

Chapter 6 describes our methodology for the construction of a rule-based face-center localization system. The goal is to locate the candidate regions in the source image that may contain face-centers.

Chapter 7 describes the face detection stage of this research. The detection stage consists of four steps: the cropper, histogram equalizer, texture-analyzer, and ANN face detector. The system uses a neural network-based classifier trained on examples of faces and non-face images.

Chapter 8 provides the conclusion and implication for future direction.

CHAPTER TWO

FACE DETECTION TECHNIQUES – LITERATURE REVIEW

2.1 Introduction

In this chapter, we review the existing techniques for automatic human face detection from colored or gray scale images. It gives a good insight into the problem and a general idea about the different approaches used in attempting to solve the problem. Then, the existing illumination enhancement methods are reviewed due to the high importance of this step to improve the detection rate. Some terminologies (face localization, facial feature detection, face recognition, face tracking and facial expression recognition) that are closely related to the face detection are defined as follows:

- **Face localization:** is to find the face location in a canonical face image (e.g. passport-like image). Face detection is a more general case of face localization. Therefore, face localization is a simplified detection problem (Yang *et al.*, 2002).

- **Facial feature detection:** finds the location of the most characteristics face components (e.g. eyes, nose, mouth, eyebrows) within images that depict human faces (J. S. Lee, Kuo, Chung, & Chen, 2007). The input may be a sub-image (i.e. window) containing only one face.

- **Face recognition:** is to identify an individual from still or video photograph image(s) of his face based on similarity of some features in an already stored face database (Li & Zhang, 2004). Face recognition can be used in two different modes:

  i) Authentication (identity verification)

  It occurs when the individual claims to be already enrolled in the system; in this case the face sample obtained from the individual is compared to the face data already stored in the database.

ii) Identification

This occurs when the identity of an individual is priori unknown. In this case, the individual's face image is matched against all the records in the face database and a match, if any, is reported.

- **Face tracking:** is a system that finds the location of a face in a sequence of images from real time video camera (Verma, Schmid, & Mikolajczyk, 2003).

- **Facial expression recognition:** is to estimate the state or mood of an individual based on his facial expressions (e.g. happy, angry, sad, surprised, fear, and disgust) (H. Y. Chen, Huang, & Fu, 2008).

It is worth mentioning that the term "face detection" is widely used in many published works, but the results and examples in these works show only passport-like input image (i.e. face localization).

An early attempt to detect frontal faces from images is dated 1969 with the work reported by Sakai Nagao and Fujibayashi (1969). They used digitized images which had eight gray levels. Standard patterns were defined in terms of line segments. The method computes the correlation values for facial features and then uses these values to detect faces. The technique was very simple and entailed various assumptions and pre-conditions.

Compared with face recognition, the topic of automatic face detection for its own sake was relatively unexplored until the early 1990s, when practical face recognition systems started to become a reality (Rowley, 1999). Over the past 20 years, the field has become more developed and extensive research has been done by researchers to develop new methods. Some methods are only good for one face per image, while the others can detect multiple faces from an image.

Also, some approaches are good for detecting faces in static images, while others are good for real-time face detection (Yang *et al.*, 2002).

There are two kinds of errors that face detectors can have:

- False Positive (FP): the system accepts an image region to be a human face, when it is not.
- False Negative (FN): in which a face in the source image is not detected. The system does not find the current face image containing a data similar enough to a face.

In practice, detecting human faces is an exhaustive search in the source image since there can be numerous possible places where a face may be found. A face may appear anywhere from the upper left to the lower right corner of the image, scaled to different sizes such as very small face or may fit almost the whole image.

Detecting human face methods are classified into four categories (Yang *et al.*, 2002), namely; 1) Feature based approaches, 2) Appearance-based methods, 3) Knowledge-based methods, and 4) Template-matching methods. Many methods may overlap category boundaries. A survey on face detection methods can be found in (Chellappa *et al.*, 1995; Hjelmas & Low, 2001; and Yang *et al.*, 2002).

2.2 Feature-based Approaches

These methods are based on the assumption that there must be some typical face properties or features that are invariant over face appearance variability such as edges, binary patterns, skin color, and facial features. The sections below cover the techniques based on edges, local binary patterns (LBP) and local gradient patterns (LGP), facial features, AdaBoost, skin color and multiple features.

2.2.1 Edge-based Techniques

Edges can be defined as locations in the source image where there is a sudden variation in the color or intensity level of pixels. The most common way for edge detection is to use a mask (also referred to as operators, filters, or templates) which is basically a small 2D array, in which the values of the coefficients determine the process to be applied such as edge detection, image sharpening, smoothing and noise removal. The mask is passed over the image at every pixel. There are different edge detection operators such as Robert, Sobol, Prewitt, Canney, etc. The Sobel operator is the regular operator, while the Canny edge detector is the latest improvement step in edge detecting approaches (Efford, 2000) and still it outperforms many of recent algorithms (Oskoei & Hu, 2010).

Craw, Ellis, and Lishman (1987) proposed a face localization method to follow the outlines of the person's head, eyes, eyebrows and mouth. Their work uses Sobel filter to find edges which are grouped together to construct a face model. The application software is designed to automate the extraction of certain measurements for mug-shot retrieval system where some kinds of constraints are imposed on the source image contents (Craw *et al.*, 1987).

Sirohey (1993) proposed a localization method based on an edge map. The system uses heuristics to assemble the edges so that the face contours are well-maintained. Then, to fit the boundary of the head region, an ellipse is used for this purpose (Figure 2.1). The detection rate was 80% on a database of 48 images.

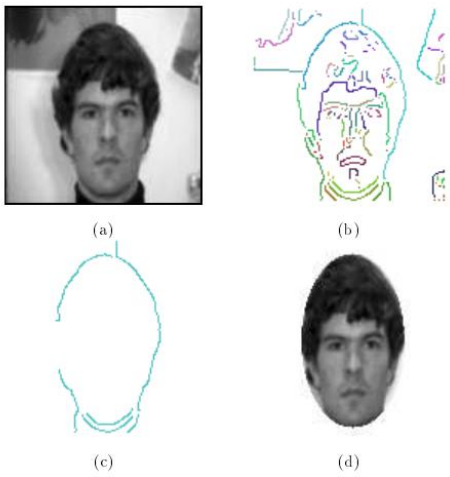


Figure 2.1: Face localization using Edge-based technique proposed by Sirohey (1993); (a) Input image; (b) edge map; (c) linked segments; (d) extracted image.

Govindaraju's (1996) edge-based algorithm involves the following steps: edge detection, thinning, removal of unwanted edges, filtering, corner detection, and edge labeling. The edges and curves are used to describe the face model. The ratio of height to width for an ideal face is used to estimate face. The author reported that the test set contained 60 images with 90 faces. The detection rate was 76% with two false alarms as average per image.

Su and Chou (1999) proposed a face detection algorithm which is similar to the work of Sirohey (1993). The edge map is generated using Sobel filter. Then, the system tries to locate ellipsoidal-shaped objects in an image. In the second stage two auto associative memories are used to validate candidate faces as faces or non-faces. The face images are of size 41×50 pixels. The system identified 32 faces correctly from a total of 35 faces.

Wang and Tan (2000) proposed a face detection system for images with the attempt to apply a special template containing directional information of edges. Initially, histogram equalization is used to enhance the source image, and then edge detection applied. The detected edges linked together using special functions called energy functions. Based on the direction information of the linked edges, the face contour finally extracted. Then a face template based on edge direction is used to detect faces. Their system assumes that the contour of a human head can be approximated by an ellipse. Figure 2.2 and Figure 2.3 show the basic flowchart of the Wang's algorithm and the results of edge linking, respectively. Experimental results show that the system can locate 87.5% of the faces with 12.5% false rate. The face detector was tested using images collected from the WWW, MIT (of Massachusetts Institute of Technology) ("MIT Face Database," 2012), and CMU databases (of Carnegie Mellon University) ("CMU Face Database," 2012).

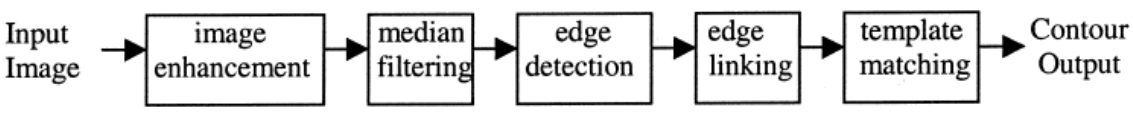


Figure 2.2: Basic flowchart of the algorithm proposed by Wang and Tan (2000).

Figure 2.3: Edge linking proposed by (J. Wang & Tan, 2000); (a) source image; (b) edge map; binary image.

Taso *et el* (2010) proposed a face detection method that is based on three techniques: edge detection, data mining, and Support Vector Machines (SVM). First, the system detects the edge-map of the facial features by applying Sobel filter, morphological operator, and thresholding. Then, the Maximal Frequent Item sets Algorithm (MAFIA) is used to mine the maximal frequent patterns from those edge images and obtain the positive feature pattern. MAFIA, which is an algorithm for mining maximal frequent item sets from a transactional database, is especially efficient when the item sets in the database are very long. MAFIA efficiently stores the transactional database as a series of *vertical* bitmaps, where each bitmap represents an item set in the database (D. Burdick, Calimlim, Flannick, Gehrke, & Yiu, 2005). Based on the feature patterns mined, a face detector based on sliding-window technique is used to search the input image at different scales. The face detector involves three cascaded classifiers where each one is used to prune certain non-face candidates. In order to enhance the performance of the face detector and avoid many of false positives FPs, a k-dimensional-tree based on SVM classifier is used for refining the final results (SVM is discussed in Section 2.3.7). Although the system uses a sliding-window of size 21×21 pixels, the classifier is trained using a feature vector that is formed from only eight rows of a raw image as shown in Figure 2.4. The figure shows the rows in range [5, 8] and [11, 14] that are used.

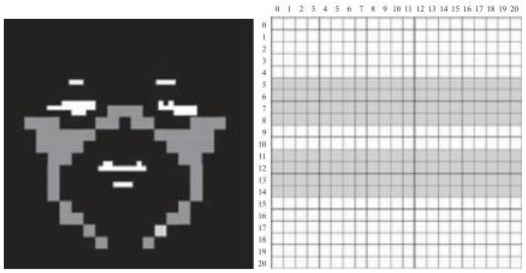


Figure 2.4: The rows used to form a feature vector, adopted by Tsao (2010).

In general, face localizing problem is not complex when dealing with uniform background images. By considering real complex images, the edge-based techniques are inadequate. Building a face model in terms of low level image features such as edges is always very difficult as the image structure changes very drastically in different images due to many random factors. Figure 2.5 shows many edges in an arbitrary image. Since real images contain many objects other than faces that may form a complex edge map, detecting faces becomes impractical for complex images and will impose high computation cost with high FP.

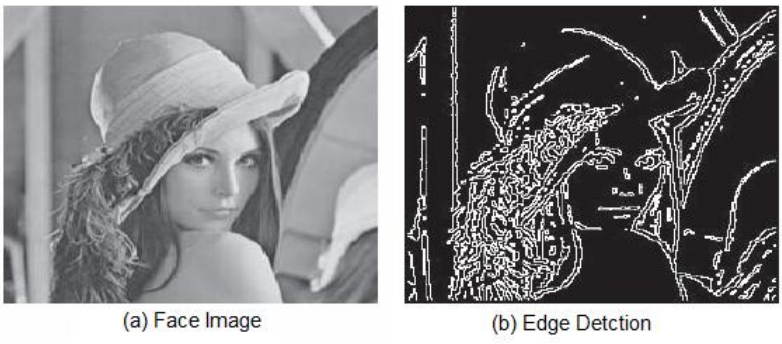


Figure 2.5: Edge-map may be changed due to variation in imaging conditions and noise; (a) Source image; (b) edge detection.

2.2.2 Local Binary Patterns (LBP) & Local Gradient Patterns (LGP)

Local Binary Pattern was used by Ojala, Pietikäinen, & Harwood (1996) for texture classification due to its invariance to global intensity variations. The idea uses a 3×3 kernel to summarize the local structure of an image. At a given pixel position (x,y), it inspects the 3×3 neighborhood pixels and generates a 1 if the neighbor pixel has a value

greater than or equal to the pixel's value, or a 0 otherwise, as indicated in Figure 2.6, while Figure 2.6(a) provides the original data and Figure 2.6(b) shows the output of this step. Then, they are multiplied by the weights given to the corresponding pixels as shown in Figure 2.6(c). Figure 2.6(d) shows the result. Finally, the sum of the eight pixels is calculated to obtain the number (169) of this texture unit.

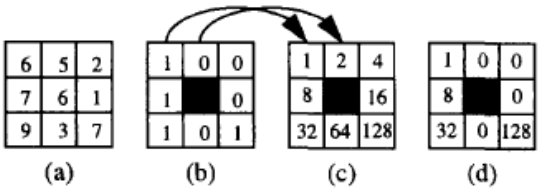


Figure 2.6: Local Binary Patterns; (a) original data; (b) thresholding; (c) weights; (d) thresholding multiplied by weights, proposed by Ojala (1996).

Many variants of the LBP have been applied. Jin *et al.* (2004) improved the LBP method for face detection. The proposed method takes the texture information and local shape into consideration. The authors reported that the proposed method is also robust to illumination variation. The face and non-face classes are modelled using multivariable Gaussian Model and classified under the Bayesian framework.

Another version was proposed by Zhang *et al.* (2007), who presented a new set of distinctive rectangle features called the Multi-block Local Binary Patterns (MB-LBP) for face detection (see Figure 2.7). Based on the MB-LBP features, a boosting-based learning classifier was developed to detect the human face. The face detector was trained based on the MB-LBP features and tested on the CMU database. The trained face detector has nine layers including 470 MB-LBP features. The MB-LBP shows a 15% higher correctness rate than the Haar-like feature and 8% higher than the original LBP features.

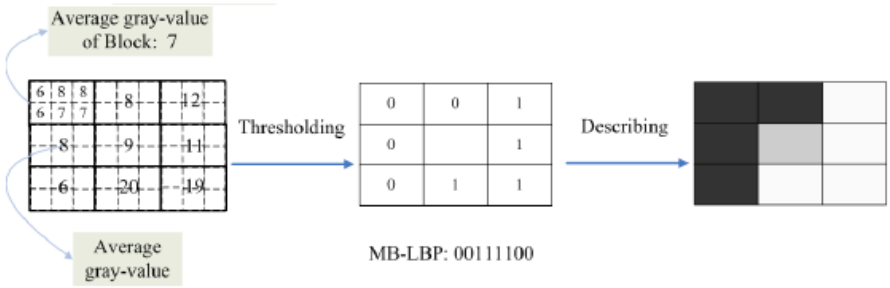


Figure 2.7: Multi-block LBP feature for image representation; proposed by Zhang (2007).

The Local Gradient Patterns (LGP) is proposed by Jun & Kim (2012), in which each pixel (x,y) is processed as follows: if the neighboring gradient of it is greater than the average of eight neighboring gradients, it is assigned the value 1; and 0 otherwise. The authors reported that the detection rate is improved by 5–27% compared with the existing methods, and the number of false positives is reduced substantially.

2.2.3 Facial Features

In addition to edge details, the pixels intensities within a human face are also used to extract facial features that can be used for face detection.

Yow and Cipolla (1997) used a human face model with six facial features as in Figure 2.8 . The system assigns probabilities to each of them, and reinforces these probabilities using Bayesian reasoning techniques. To deal with missing features, the face model is further decomposed into components consisting of four features. Then, the model is further subdivided into components consisting of two features. Facial features that are invariant to changes in scale and illumination intensity are also used (see Figure 2.9).

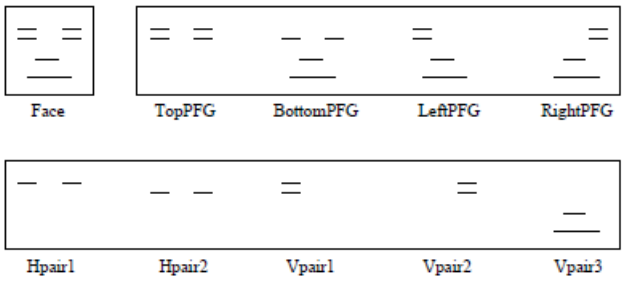


Figure 2.8: The face model and its components proposed by Yow and Cipolla (1996).

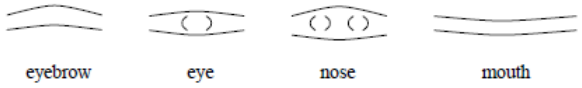


Figure 2.9: The facial feature models proposed by Yow and Cipolla (1996).

At the first stage, a list of interest points is found from the image using a second derivative Gaussian filter, and then local maxima and edges are searched. Then, Attentive feature grouping process is used as shown in Figure 2.10. The authors reported 85% detection rate using 110 images with a false detection rate of 28%.

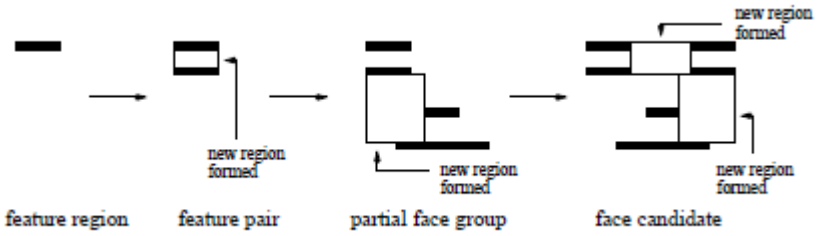


Figure 2.10: Attentive feature grouping, proposed by Yow and Cipolla (1996).

Jeng *et al.* (1998) proposed a geometrical face model based on facial feature detection. Each facial feature has an associated evaluation function, which is used to determine the final most likely face candidate. The weights of the facial features are selected experimentally and correspond to their importance as shown in Eq. (2.1).

$$E=0.5E_{eye} + 0.2E_{mouth} + 0.1E_{Righteyebrow} + 0.1 E_{Lefteyebrow} + 0.1E_{nose} \quad (2.1)$$

The reported detection rate is 86% on a dataset of 114 test images.

Sinha's (2002) approach is based on the observation that qualitative darkness relationships between facial features are invariants to changes in lighting conditions. So, a set of relationships between facial features would be adequate to detect a face. Accordingly, the face template is divided into a set of regions and a set of relations as shown in Figure 2.11. Each arrow in the figure refers to a relation that is used along with a predefined threshold value to find whether the relation is satisfied (i.e. it is above the threshold value). A face is detected if an image window satisfies all the relations.

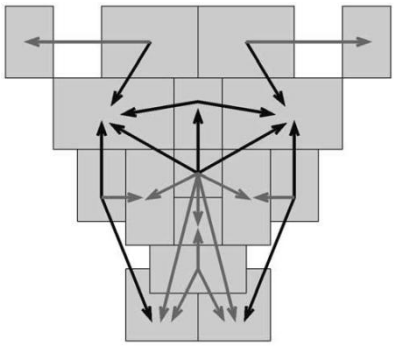


Figure 2.11: The face template is composed of a set of regions and a set of relations. The template size is 14×16 pixels; proposed by Sinha (2002).

The main drawbacks of facial features-based approaches are:

- Such approaches are infeasible to detect faces in an unconstrained environment due to features extraction errors. The image features can be severely corrupted due to illumination, noise, and occlusion. Feature boundaries can be weakened for faces, while shadows can cause numerous strong false edges.
- Building a statistical model to describe the features and relationships is not an easy task and would be based mainly on training samples.

2.2.4 AdaBoost-based methods

Viola and Jones (2004) introduced the "Integral Image" which is another form of image representation used to quickly calculate responses of a set of image-based features (i.e. Haar-like features). Therefore, the approach does not work directly with image intensities data. The integral images can be defined as two-dimensional matrix that holds global image information

where each element at location x, y contains the sum of the pixels above and to the left of x, y, inclusive:

$$ii(x,y) = \sum_{x' \le x, y' \le y} i(x', y') \tag{2.2}$$

where $ii(x,y)$ is the integral image and $i(x,y)$ is the original image. Using recurrences, the integral image is quickly computed in a single image pass (Sonka *et al.*, 2008). The number of features that can be calculated is quite large. Consequently, a subset of best features needs to be identified. A learning algorithm based on a modified AdaBoost algorithm is employed to select a small number of well distinguishing features. The feature selection is performed in such a way that each weak classifier depends on a single feature. A set of weak classifiers are combined in a "cascade" sequence. The early-stage weak classifiers are set so that their false negative FNs are close to zero. Minimizing FNs of course increases the false positive FPs. Ultimately; the remaining non-rejected locations are marked as the locations of identified faces. For instance, at the first stage two simple filters are used, each composed of a few rectangular light or dark regions as shown in Figure 2.12. This stage can detect 100% of the faces with about 50% FPs. The subsequent classifiers are called until all requirements for the detection performance are met. Like most face detectors, the systems scan the input image many times at many scales (i.e. 12 scales). Faces are detected at a size of 24×24 pixels. The complete system consists of 38 stages with over 80,000 operations. Nevertheless, the cascade structure results in extremely rapid average detection time compared to any previous approach. The system was tested on the MIT + CMU frontal face test set. The dataset consists of 130 test images containing 507 faces. The reported detection rate is 76.1% to 94.1% with 10 to 422 false detections respectively.

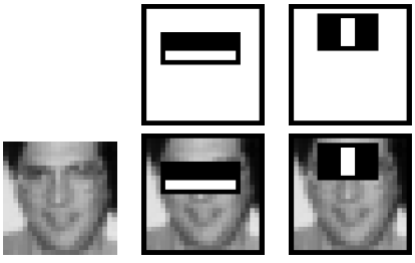


Figure 2.12: Sample of the features proposed by Viola and Jones (2004).

Boosting weak classifiers has been shown in many works such as those by Fleuret and Geman (2001), Li and Zhang (2004), Wu, Ai, Huang, and Lao (2004) Xiao, Li, and Zhang (2004), Ding and Martinez (2010), and many others.

For the multi-view face detection, Wang and Ji (2007) presented a statistical learning method to extract features. The authors argued that the Recursive Nonparametric Discriminant Analysis (RNDA) can handle more inter-class variations. Histograms of extracted features are learned to represent class distributions and to construct probabilistic classifiers. Then, the features are learned and combined with AdaBoost to form an efficient face detector. During training, all collected face images are normalized to a fixed size of 20 × 20 pixels. The reported detection rate is 84.5% and 90.2% for frontal face images from CMU+MIT databases respectively.

A new feature selector and learning strategy for AdaBoost was proposed by Wu *et al.* (2008). The authors reported that a Forward Feature Selection (FFS) algorithm is faster than a naive implementation of the AdaBoost feature selection method and can achieve similar detection accuracy. The reported false detection rate was reduced by 36.1% compared with the naive implementation of AdaBoost.

During the training of AdaBoost cascade classifiers, Xiaohua *et al.* (2009) found that the level of complexity increases quickly, while the error rate of non-face examples decreases slowly with an increase in node depth. Therefore, the authors proposed to boost the cascade architecture based on a hierarchical strategy using information from the surrounding objects of the face regions step by step; and they also used simplified Gabor features to extend the feature set for the training of deeper nodes. The reported experiments show that the detection performance was improved by about 10% when compared to the original AdaBoost method.

Guo *et al. (2010)* (2011) used Pixel-Based Hierarchical-Feature Adaboosting (PBHFA) method along with Probability-based Face Mask Pre-Filtering (PFMPF) to reduce the training time of traditional Adaboosting. The authors reported a detection rate of 90.7% and 98.06% for test images from CMU and FERET databases respectively.

Belaroussi and Milgram (2012) did a comparative study on four boosted classifiers. These classifiers are used for face detection and tracking algorithms and compared in terms of speed and efficiency. The authors reported that the Discrete Adaboost classifier shows best results with an input image size of 24×24 pixels.

Researchers in face detection applications usually aim at high speed performance in order to make their systems practical for real-time system and embedded consumer applications. Yang, *et al.* (2010) proposed an AdaBoost-based face detection for embedded systems. The system was implemented using Field-Programmable Gate Arrays (FPGAs). The Field-Programmable Gate Array (FPGA) is a semiconductor device that can be programmed after manufacturing, which is used as a component for high performance computing. The authors also proposed to use Genetic Algorithm (GA) in the AdaBoost training to minimize the false positive rate given the number of Haar features; while in the detection stage, for a hard real-time design, they proposed a new complexity control scheme in which unlikely candidate windows are skipped based on spatial correlation between successive scales. The simulation results show a detection rate of about 75–80%.

Ding *et al.* (2012) later proposed a face detection system which is also implemented using FPGAs and based on Haar-like features as weak classifiers. The authors presented the implementation to improve the integral image pipeline calculation design. The detection rate is 87.2% with high processing speed up to 100 fps (frames per second).

A main drawback of Adaboost is that the large number of possible Haar-like features in a standard sub-window may be time consuming, which makes specific environment feature

adaptation extremely difficult. For example, Viola and Jones (2001) reported that about 45,396 rectangle features are used with the pattern of size 24 × 24 pixels.

2.2.5 Skin Color

Color has been used and proven to be a key feature for object detection. Color information under constrained lighting conditions is robust against changes in scale, position, rotation and partial occlusion of the objects in images (Moallem et al., 2011). Therefore, detection human skin regions in images based on skin color information is an efficient approach for face detection. Moreover, the processing of color information has proven to be much faster than processing of other facial features (Gonzalez *et al.*, 2007).

Skin color modeling and detection methods are described in Chapter 3.

2.2.6 Multiple Features

These methods locate or detect faces by combining multiple features. Zaqout, Zainuddin, and Baba (2004) developed a two-phased face detection system in colored images. First, it applies skin color segmentation to detect skin-like regions based on lookup-tables in RGB color space. In the second stage, the system detects faces based on the assumption that the appearance of face is blob-like and has an approximately elliptical shape. The detection rate of skin region in the test images using Compaq database is about 95.17% with about 17.31% False Positives.

(Juang & Shiu, 2008) have proposed a three-stage face detection method. The first two stages are similar to Zaqout's approach (2004). In the final stage, self-organizing fuzzy network with support vector learning is used to make the final decision which is based on the extracted facial features such as colors of the eyes and mouth. The reported detection rate is 95.4-95.7% with 60-63 false detections respectively. The dataset consists of 400 + 350 images containing 456 and 350 faces respectively

Zhao-Yi, Yu, and Ping (2009) also used adaptive skin color along with face structure models to detect faces. The detection rate on a test set of 100 images containing 202 faces is 195 with 3 false detections.

2.3 Appearance-Based Methods

Recently, increasing activities have been noticed in developing approaches to detect faces under unconstrained scene conditions. Among the many face detection methods, the appearance-based methods have been gaining more attention. These methods rely on techniques from machine learning and statistical analysis. The appearance-based methods process the input image without image contents analysis or features extraction. Only low-level image processing such as image enhancement and lighting correction are necessary.

These methods are based on building a classifier which processes fixed-size images, and determines whether a given subimage window correspond to a face or not. Generally, a sliding window technique is used for detecting faces. There are variations in the implementation of these methods such as the size of the sliding window (i.e. face pattern), the subsampling degree, the size of each step, etc. These methods include Principle Component Analysis (PCA), Neural Networks (NN), Support Vector Machines (SVM), Bayes Classifier, Hidden Markov Model (HMM), and so on.

2.3.1 Principle Component Analysis (PCA) or Eigenfaces

In most applications, particularly in object detection and recognition, there is a need for dimensionality reduction of feature vector. Principal Component Analysis (PCA) is a statistical technique commonly used in dimensionality reduction and extraction of statistically independent features that has found application in fields such as face detection, face recognition, and image compression (Smith, 2002). Depending on the field of application, PCA is also named the discrete Karhunen-Loève transform (KLT), or the Hotelling transform (Jolliffe, 2005).

An early example of employing PCA to represent human faces was done by (Kirby & Sirovich, 1990). The technique first finds the principal components of the distribution of faces, expressed in terms of eigenvectors. Each face in the training set can be represented as a linear combination of the best *K* eigenvectors, which are referred to as "*eigenpictures*" or "*eigenfaces"*, because of their face-like appearance.

Turk and Pentland (1991a) later developed this technique for face detection and recognition. The developed procedure for computing the face space is as follows:

1. The training face images must be centered and of the same size, Figure 2.13(a).
2. Represent every image as a vector.
3. Compute the average (mean) face.
4. The average face is then subtracted from each image.
5. Compute the covariance matrix.
6. Compute the eigenvectors and eigenvalues of the covariance matrix.
7. Order the eigenvectors by eigenvalue, highest to lowest. This gives us the components in order of significance. Keep only *K* eigenvectors (corresponding to the *K* largest eigenvalues). Here is where data compression and reduced dimensionality come into play (we can ignore the components of lesser significance).

The projection of an image into eigenspace will transform the image into a representation of a lower dimension aims to hold the most important features of the face and make the comparison of images easier (Nes, 2003). Figure 2.13(b) shows an example of eigenfaces that could be generated to make up the face space.

To detect the presence of a face in a specific image region (i.e. sliding window), the system computes the distance between an image region and the face space. This distance, referred to as

Distance From Face Space (DFFS), is used as a measure of "faceness". If the distance is less than threshold *T* then the image window is a face.

Figure 2.13: Eigenfaces; (a) sample of 40 training faces; (b) eigenfaces; adopted from (Johnson, 2012).

Pentland, Moghaddam, and Starner (1994) proposed the "eigenfeatures" instead of eigenfaces (i.e. eigeneyes, eigennose, eigenmouth). These eigenfeatures were calculated from the training facial feature templates. The authors argued that the eigenspace formulation leads to a powerful alternative to simple template matching. Later, Moghaddam and Pentland (1997) extended their work for target detection in application to human faces and hands. The reported detection rate for a set of 7000 faces is 95% to 97%.

(Samal & Iyengar, 1995) used PCA to obtained a set of basic face silhouettes. Based on the face silhouettes, the authors developed a system that performs the task of human face detection automatically. Edge detection, thinning, thresholding, and other image processing techniques are also used. The reported detection rate is 92% using 129 test images.

Chen and Lien (2009) proposed a statistical-based system for automatic multi-view face detection. The system uses PCA and independent component analysis (ICA) coefficient vector, respectively. These vectors are modeled using the Gaussian mixture model (GMM). The detection is based on the calculated probability of the weight and coefficient vectors. The authors reported that the system shows over 90% accuracy rate.

2.3.2 Factor Analysis

Factor analysis (FA) is a statistical model which is similar to PCA in many aspects (Yang *et al.*, 2002). FA is better than PCA from the view point that FA defines better density model for the data and it is more robust to independent noise (M. H. Yang, Kriegman, & Ahuja, 2001). Yang *et al.* (2001) proposed a mixed model of factor analyzers and performed clustering of face patterns. Then, dimensionality reduction is done for each cluster. To detect faces, a sliding window is passed over the input image. For each subimage window, the system calculates the probability of being a face. If the probability is above a predefined threshold, then a face is detected. The detection rate reported is 92.3% with 82 false positives, where the test images contain 483 faces in a set of 125 images.

2.3.3 Point-Distribution Methods

With high diversity of face appearance, methods have been proposed based on mixture of multi-dimensional Gaussians (Sung & Poggio, 1998). The system comprises four steps. First, the system resizes the sub-image window into 19×19 pixels, eliminates near-boundary pixels, and applies histogram equalization. Then, the face and non-face training images are grouped into twelve multi-dimensional Gaussian clusters. Six clusters correspond to face and the other six clusters correspond to non-face clusters (see Figure 2.14). These clusters are constructed by an elliptical k-means clustering algorithm. For new image, two distance metrics are computed relative to the canonical face model, namely: Euclidean distance and Mahalanobis distance. Then, twelve pairs (i.e. 24) of distances are used to train a multi-layer perceptron network (MLP) to classify an image window. The training samples consist of 47,316 images, where 4,150 are face samples.

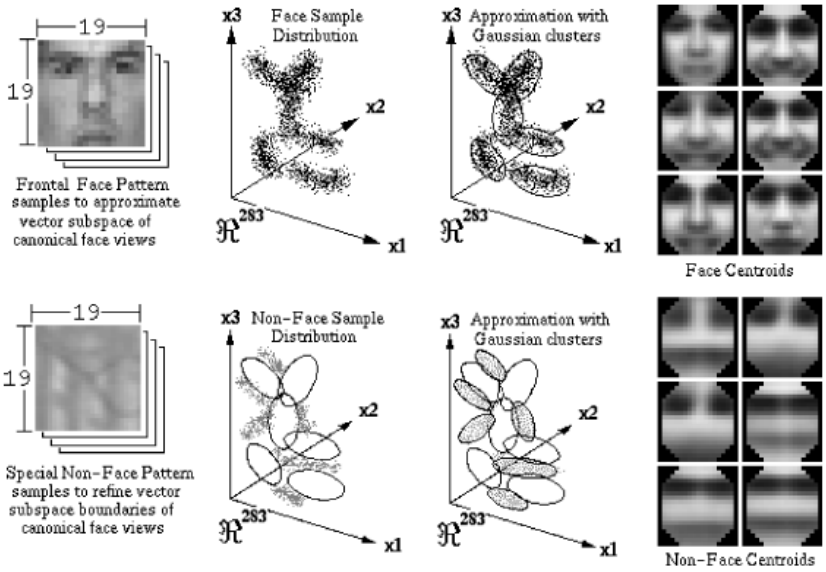


Figure 2.14: Point distribution-based method proposed by Sung and Poggio (1998). Top row: the distribution of face patterns using six Gaussian clusters. Bottom row: The final model consists of 12 Gaussian clusters; six "face" clusters and six "non-face" clusters.

2.3.4 Artificial Neural Networks (ANNs)

Since we used a neural network-based classifier as a part of our system (see Chapter 7), this section presents a description of neural networks model, learning methods, and then a review of the existing neural network-based face detectors.

2.3.4.1 Introduction to ANNs

The word *neural* refers to the fundamental unit in neural networks which is the Neuron. Neurons have the ability to learn and thereby acquire knowledge and making it available for use. The word *network* refers to the inter–connections among the neurons. All the neurons with their connections constitute the *artificial neural network (ANN)*. It is a parallel distributed processing system that attempts to emulate electronically the architecture and information representation scheme of the biological model of the human nerve system (Pal & Mitra, 1999; Rajasekaran & Pai, 2004). The human nerve system is capable of processing and analyzing a massive amount of complex information. It is estimated that a human has a total of about 10^{11} neurons (Hagan, Demuth, & Beale, 1996; Haykin, 2009; Kinnebrock, 1995; Mitchell, 1997).

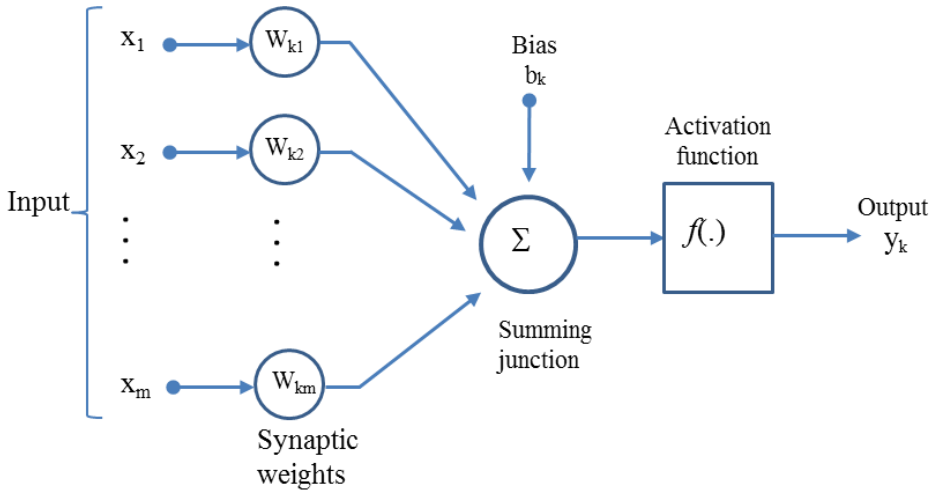
Every neuron has an average of 10,000 connections to its neighbors. Neurons are the actual processing units in the brain and like any processor they possess an input and output.

An early milestone literature that is related to the ANNs can be traced to a simple neuron model introduced by McCulloch and Pitts (1943), as stated in (Gonzalez & Woods, 2002; Haykin 2005; Pandya & Macy, 1996). The progress of neurobiology has boosted the efforts to build simple mathematical models that are inspired by biological neural networks. According to (Krose & Smagt, 1996), when Minsky and Papert published their book *Perceptrons* in 1969 (Minsky & Papert, 1969) that revealed the deficiencies of perceptron models, most researchers left the field (Krose & Smagt, 1996). Other reasons such as lack of ideas and powerful computers also lessened interest in ANNs (Haykin 2005). Only a few researchers continued their efforts (Krose & Smagt, 1996). Interest in neural networks re-emerged in the early 1980s after important theoretical results were achieved such as the development of error back-propagation and the hardware advances (Haykin, 2009).

An important characteristic of the ANNs is the capability to *learn* from input data (Haykin, 2009). Since face detection problem can be regarded as a two-class classification problem, that is an image window is either a face or non-face image, the ANNs can be trained for the classification task.

2.3.4.2 Artificial Neural Network Model

In general, each neuron receives a vector of scalars (X) that is multiplied by a vector of weights (W). The result of adding a bias value (b) to the product of vectors $(X.W)$ will be applied to activate the function (f), which is called the transfer function. A variety of transfer functions in an ANN can be employed. Figure 2.15 shows a typical diagram of a neuron (Haykin, 2009).



$$y_k = f(W_k.X + b_k)$$

Figure 2.15: Input and output of an ANN neuron, adopted from (Haykin, 2009).

Neurons in an ANN are distributed on three types of layers: input layer, hidden layer(s), and output layer. There are several types of architecture of ANNs. However, the most widely used ANNs are Feed Forward Networks and Feedback/Recurrent Networks (Munakata, 2008). A number of methods are used to initialize and adjust the ANN weights. One way is to set and initialize the weight values depending on prior knowledge. Another way is to initialize them with random values and then train the ANN by feeding it training patterns and forcing it to change its weights according to the learning rules. The methods for training ANNs can be classified into two distinct categories: supervised and unsupervised learning. Both learning methods result in the adjustment of weight values between units. It is estimated that 80-90% of neural network applications is developed using Feed Forward Back propagation Neural Network (FFBP-NN) algorithm or its derivatives (Munakata, 2008). A standard FFBP-NN is a gradient descent algorithm in which the network weights are moved along the negative of the gradient of the performance function (see APPENDIX-B). There are a number of variations on the basic algorithm that is based on other standard optimization techniques, such as conjugate gradient and quasi-Newton methods (Haykin, 2009).

2.3.4.3 ANN-Based Face Detectors - Background

An early hierarchical neural network for face detection was reported by (Agui, Kokubo, Nagahashi, & Nagao, 1992). In this method, a picture of human faces is limited to have almost the same size. The system consists of two stages. In the first stage, two parallel sub-networks are used, in which the inputs are the pixels intensities of original image and pixels intensities from filtered image using a Sobel filter of size 3 × 3. Then, some features are calculated (extracted) such as the standard deviation and some geometric moments. These extracted features are passed to the second stage network which will decide whether a face is present in the input region or not. The run time for locating faces in an image of size 512× 512 is about 60-120 seconds run on SUN-4 System.

Propp and Samal (1992) developed a neural network-based face detection. Their network is designed with 1,024 input units and consists of four layers. The three hidden layers contain 256, eight, and two units respectively as cited by (Yang *et al.*, 2002).

Soulie, Viennet , and Lamy (1993) proposed a time-delay neural network with sliding window of size 20×25 pixels that passed over the input image. The wavelet transform is used to decompose the image. They reported a false negative rate of 2.7% and false positive rate of 0.5% from a test of 120 images.

In the work of Vaillant, Monrocq, and Le Cun (1994) a convolutional neural network is applied for localizing faces in images. The system consists of two steps. First, a neural network is trained to find approximate locations of face candidate areas. Then, these areas are passed to another neural network to find the precise position of the face. The training face and non-face images are of size 20×20 pixels. The reported detection rate is 90-96% with 181-635 false detections respectively.

Probabilistic Decision-Based Neural Network (PDBNN) was proposed by Lin , Kung, and Lin (1997) for detection/recognition of human face. The system used pattern size of 12 × 12 pixels. To detect a face in the source image, the system searches all possible sub-images and a confidence score is produced by the neural network. If the confidence score is above some threshold value, a face is detected. The author stated that the experiments were performed on about 1000 testing patterns and the proposed system has consistently and reliably determined actual face locations.

From our point of view, the most significant work based on ANNs is that by (Rowley, 1999; Rowley *et al.*, 1998). A multilayer neural network is used to learn the face and non-face patterns from face/non-face images. The system uses a sliding window of size 20×20 pixels region. The network has retinal connections to its input layer; there is one hidden layer with 26 units divided into three types of hidden units, where 4 hidden units are connected to 10 x 10 pixel sub-regions, 16 units connected to 5 x 5 sub-regions, and six units connected to 20 x 5 pixels via input units. The input window is pre-processed through lighting correction and histogram equalization. The system was tested on two large sets of images collected from CMU+FERET datasets. The reported detection rates using different ANN structures show that the best acceptable trade-offs between the number of false detections and the detection rate is by using only two networks which is based on *Thresholding* and ANDing arbitration strategy. This structure detects on average 86.2% of the faces with an average of one false detection per 3,613,009 windows examined.

Chang, Hung, and Lee (2008) proposed to use a specific optimization technique named DIRECT to develop a neural network face detector based on Rowley's approach (1998).

Curran, Li, and Mc Caughley (2005) proposed a neural network-based face detector. The size of face pattern is 18 × 27 pixels and ignores the corner pixels. The system used a multi-layer network with 400 input variables (of each training image), 25 hidden units, and one output unit.

The algorithm can detect between 67% and 85% of faces from images under unconstrained conditions with an acceptable number of false detections.

Smach, Atri, Mitéran, and Abid (2006) designed a neural network classifier. The proposed classifier shifted the detection stage to be in hardware implementation instead of software allowing real-time processing. No quantitative results are given.

The main disadvantages of ANNs:

- The ANN-based classifier needs training to operate.
- The architecture of a neural network is different from one designer to another even for the same problem and has to be tuned by extensive experiments to get the best results.
- The processing time for large neural networks may be high.

2.3.5 Sparse Network of Winnows

The Sparse Network of Winnows (SNoW) was used by Yang *et al.* (2000) to detect human faces from gray scale images. The author argued that SNoW learning architecture is specifically designed for learning in the presence of a very large number of features. The system used 1681 face and 8422 non-face images for training. The pattern size is 20× 20 pixels. The system was tested on two sets of images collected from CMU and (Sung & Poggio, 1998) datasets. The first dataset consists of 130 images with 507 faces and the second dataset consists of 23 images with 155 faces. The reported detection rate is 93.6% to 94.8% with false detections of 3 and 78 respectively.

2.3.6 Wavelet Transform

The *transform* of a signal (a vector or an image) is a *new representation* of that signal. An early milestone literature that is related to the wavelet transform can be traced to the mathematician Alfred Haar in 1909, as stated by (Stollnitz, DeRose, & Salesin, 1996). However, the concept of

the wavelet did not exist at that time. Afterward, Grossman and Morlet invented the term wavelets in 1984 (Grossmann & Morlet, 1984).

In 1987, wavelets were first shown to be the foundation of powerful new approach to signal processing and analysis called multi-resolution theory (Gonzalez & Woods, 2002; Mallat, 1987, 1989). As its name implies, multi-resolution theory is concerned with the representation and analysis of signal or images at more than one resolution. The principal idea behind this approach is obvious - features that might go undetected at one resolution may be easily detected at another. For 2D-image implementation, this implies representing image as an image pyramid. In 1988, Daubechies constructed a family of easily implemented and easily invertible wavelet transforms (Daubechies, 1992). In 1989, Mallat proposed the fast wavelet transform. Beyond 1989, exponential growth is observed in theoretical developments (Mallat, 1989).

Wavelet decomposition has several desirable properties for pattern classification (Zhu, 2005). First, it exploits the redundancy and effectively characterizes image signals. An image pattern is decomposed into a number of sub-band signals. Each sub-band signal carries the image information of a certain orientation and frequency scale. The original image is represented by sub-images that carry a certain orientation and frequency scale of the original image.

Garcia and Tziritas (1999) proposed a face detection system based on skin color and wavelet transform to characterize the texture of a human face. The texture of each face candidate region is analyzed by performing wavelet packet decomposition and then described by band filtered images containing wavelet coefficients (see Figure 2.16). From these coefficients, a set of statistical data is extracted in order to form a vector of face descriptors. Then, the Bhattacharyya distance is used to classify the feature vector into face or non-face.

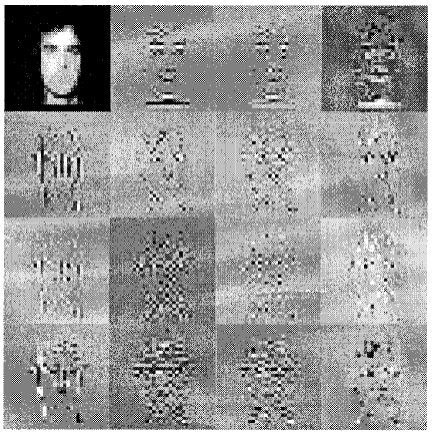


Figure 2.16: Wavelet decomposition of facial image with level 2 coefficients, adopted from Garcia (1999).

Later, Wang and Yuan (2001) used wavelet decomposition to extract facial features from candidate skin regions. Then ANN is used for eyes detection. When the eyes are detected, the region is classified as a face. The reported detection rate is 91.1% with 6.67% false detections. The dataset consists of 65 test images containing 95 faces.

2.3.7 Support Vector Machines

Support Vector Machines (SVMs) are a relatively new learning method successfully used for pattern classification such as handwritten recognition, object recognition, speaker identification, face detection, etc. (Burges, 1998). Given a set of training examples, each marked as belonging to one of two categories, the SVM training algorithm builds a model that assigns new examples to one category or the other. An SVM classifier constructs a hyperplane or a set of hyperplanes in a high dimensional space, which can be used for classification. This optimal hyperplane(s) is defined by a weighted combination of a small subset of the training vectors, called support vectors. Estimating the optimal hyperplane is equivalent to solving a linearly constrained quadratic programming problem (Osuna, Freund, & Girosi, 1997; Yang et al., 2002).

An early attempt to use SVMs for face detection is reported by (Osuna et al., 1997). An SVM is trained using two sets of face and non-face images of size 19×19 pixels. After training the SVM, the system detects faces by exhaustively scanning the input image for face-like patterns at

several scales. Other tasks such as masking, illumination correction, and histogram equalization are used as preprocessing steps to enhance the quality of the scanned subimage windows. The system was tested using two sets of images. The set A, consists of 313 single face high-quality images. The set B, consists of 23 images of mixed quality containing 155 faces. The reported detection rate is 74.2% to 97% with 20 and 4 false positives respectively.

Heisele, Serre, Prentice, and Poggio (2003) presented a SVM-based face detector in gray images. A hierarchical structure of five SVM classifiers is used with increasing complexity in each level. At the bottom and middle levels, the system uses fast classifiers to reject large regions of the input image. The slow classifier on the top level performs the final decision. The reported detection rate is 80% using CMU dataset. The authors reported that their system is faster by a factor of 335 compared to the naive implementation.

Taso *et el.* (2010) proposed a face detection method that is based on three techniques: edge detection, data mining, and SVM. This work is already discussed in Section 2.2.1.

However, the main drawbacks of SVMs are intensive requirements of the computation and memory (Yang *et al.*, 2002). To give an idea of some memory requirements, Osuna *et al.* (1997) stated - for an application that involves 50,000 training samples, and this amounts to a quadratic form whose matrix has 2.5×10^9 entries, would need 20 Gigabytes of memory (using an 8-bytes floating point representation).

2.3.8 Hidden Markov Model (HMM)

It is often possible to model the patterns being observed as a transitionary system. If the transitions are well understood, and we know the system state at certain instant, they can be used to assist in determining the state at a subsequent point. This is the idea of Markov's model. (Sonka *et al.*, 2008). Hidden Markov Models provide a high level of flexibility for modelling the structure of an observation sequence (Marchand-Maillet & Merialdo, 1999). Since facial

features regions appear with the same structure and order from top to bottom, the face image can be modelled using HMM by assigning each of these regions to a state.

Marchand-Maillet and Merialdo (1999) presented a HMM-based face detector. The idea is to exploit the vertical structure of a human face. Accordingly, face image is divided into a sequence of observation vectors. Then, modelling is done at the line level. In an image containing a face, two types of lines are distinguished, namely, lines composed of background pixels only and lines composed of a sequence of background and face pixels as shown in Figure 2.17(a). Two 1D-HMM lines are used for modelling as shown in Figure 2.17(b). As illustrated in this figure, for each state of the model there is an associated output probability which represents the probabilistic transitions between states. After training the HMM, the output probability of an observation determines the class to which it belongs. No quantitative results are provided.

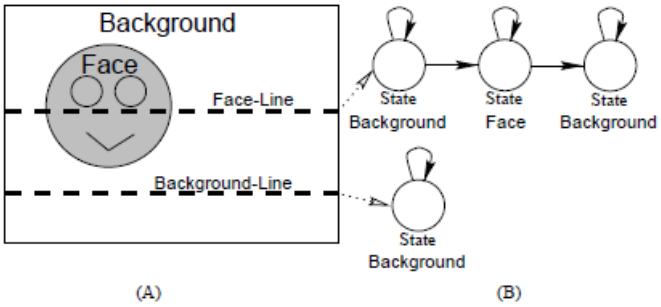


Figure 2.17: Face modeling using HMM; (a) A typical face image; (b) its model using 1D-HMM; proposed by Marchand-Maillet and Merialdo (1999)

Nefian (1999) generalized 1D-HMM to 2D structure and referred to it as an embedded HMM. The proposed method is used for face recognition and detection. In this approach, the author allows for each state in 1D-HMM to be an HMM. In this way, The HMM consists of a set of super states along with a set of embedded states. For this reason, the author called his proposed method an embedded HMM. On the MIT database, which contains 432 images each with a single face, the reported detection rate is 91.2% to 91.5% with 56 and 43 false positives respectively.

Rajagopalan *et al.* (1998) proposed a method for finding faces using HMM in a different way. Instead of using HMM to exploit the structure of a human face as in (Marchand-Maillet & Merialdo, 1999) and (Nefian, 1999), in this method the face to non-face and non-face to face transitions are learnt using a HMM. The proposed methods consist of two schemes. The first scheme estimates the density of the face and the face-like regions using higher order statistics (HOS). Then, the face to non-face and non-face to face transitions are learnt using HMM. The size of the face pattern used is 13×13 pixels. The experiments show a detection accuracy rate of 94.28% with 17 false positives on a set of images from CMU + FERET datasets.

HMM had been used for face recognition in many works such as (Samaria & Young, 1994), (Salah *et al.*, 2007), (X. Liu & Cheng, 2003), and (Bicego, Castellani, & Murino, 2003).

2.3.9 Naive Bayes Classifier

Schneiderman and Kanade (1998) argued that some local patterns such as the intensity patterns around the eyes of a human face are much more unique than the intensity patterns found on the cheeks. In order to represent the "uniqueness" of local appearance, the statistics of local appearance in the world at large should be modelled. Accordingly, the space of local appearance is partitioned into a finite number of patterns. The discrete nature of this representation allows an estimation of the statistical model P(object|image) by counting the frequency of occurrence of these patterns over various sets of "training" images. The authors used approximately 106 patterns that represent discrete representation of local appearance. Using Bayes theorem, a Bayes classifier is described to estimate the joint probability of local appearance of sub-regions of the face at multiple resolutions. This method achieves a detection rate of 86.6% with 79 false positives using a test set of (Rowley *et al.*, 1998) (i.e. 125 images containing 483 faces).

Later, Verma *et al.* (2003) developed a face detection and tracking in a video which is based on Schneiderman and Kanade (1998) work.

Liu (2003) proposed a face detection method based on Bayesian Discriminating Features (BDF). The system first derives the feature vector by combining the input image, its 1D Harr wavelet representation, and its amplitude projections. Then, the system computes the conditional Probability Density Functions (PDFs) of the face and non-face classes using 600 training images. Finally, the Bayes classifier is applied to detect frontal faces in an image. The reported detection rate is 98.5% with single false detection. The dataset consists of 887 images containing 1,034 faces.

2.4 Knowledge-Based Methods

These rule-based methods encode human knowledge of what constitutes a typical face. In general, facial features extraction should be done first. The search starts with primitive facial feature blobs and successively applies the rules which will guide the process until a total description of a face is obtained. A face candidate is identified as a face when a total description is obtained.

An early research in recognition of human faces from mugshot images is described by Kanade (1973). In order to locate facial features, the image was converted to an edge image, and then projected onto the horizontal and vertical axes. By looking for particular patterns of peaks and valleys in these projections, the program could recognize the locations of the eyes, nose, mouth, borders of the face, and the hair line. Rules are used to validate if the patterns match the expected values. If the patterns do not match the expected values then the image is rejected as a non-face. Although the research goal here is on recognition, this marks one of the first attempts to detect whether a face is present in an image.

Yang and Huang (1994) explored the intensity-scale behavior of faces in mosaic (pyramid) images. The researchers observed that small features of the face will gradually disappear when the resolution of a face image is subsampled to smaller sizes repeatedly. At low resolution, face region will turn out to be unvarying. Based on this observation, many rules and feature detectors

are used. There are three levels of rules. The higher two levels are based on mosaic images at different resolutions. The third level uses an improved edge detection method to detect edge-map. At the first level, simple rules such as "the eyes should be darker than the rest of the face" are used. All windows that pass first level are evaluated by rules at the second level, in which more detailed rules are used with higher resolution. Finally, at the third level, the edge-based rules are used to classify a window as either a face or a non-face. The authors used 60 images for testing the system, each containing one face. The system detects 50 of these faces correctly while there are 28 false detections.

Huang, Gutta, and Wechsler (1996) proposed a decision trees (DT) algorithm for detecting faces. The algorithm involves a set of rules in three stages. First, the possible location of the face is roughly estimated using techniques such as histogram equalization, edge detection, and projection profiles analysis. The second stage rules are used to refine it based on facial features. The last stage decides whether a face is present in the image. The reported detection rate is 92-96%. The test dataset consists of 2,340 faces images with semi-uniform background.

Kotropoulos and Pitas (1997) presented a rule-based localization method which extends the work of Yang and Huang (1994). First, the system computes the horizontal profile of an input image. Then, it determines any abrupt changes to find the facial features. Similarly, the vertical profile is calculated to locate the candidate facial features, Figure 2.18. Subsequently, a set of rules are used to validate these candidates. The proposed method was tested using single face images with a uniform background. These images are collected from the M2VTS database (Multi Modal Verification for Teleservices and Security applications) which contains video sequences of 37 different people. The localization accuracy is 86.5%. In general, this method cannot be applied to detect faces from complex images due to the fact that it is difficult for profile projections to locate the facial features in complex images.

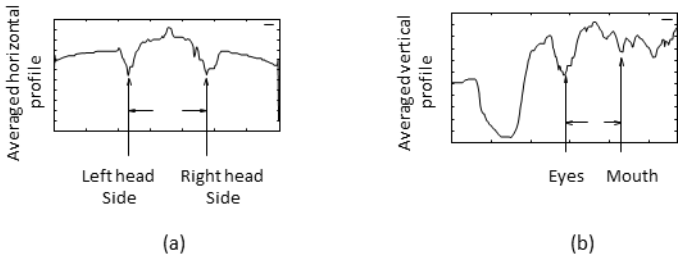


Figure 2.18: Horizontal/ Vertical profile;
(a) Horizontal profile; (b) Vertical profile; adopted from Kotropoulos (1997).

Moallem *et al.* (2011) proposed a fuzzy rule-based system to detect human faces. In this system many features such as the skin-color, position of the lips, shape information and ear texture properties are fed to the fuzzy rule-based face detector. The system works in several stages. The skin detection approach is presented in more detailed in Section 3.7.2.4. The reported detection rate is 90-98% using 783 images.

The main drawbacks of this approach:

- With diversity of images types and sources, translating human knowledge into well-defined rules is not an easy task and based on human judgment when driving these rules.
- Rules about facial features that describe human face work well with frontal faces in uncluttered scenes (i.e. uniform background). Complex images usually involve a huge amount of information at different levels. These methods will fail in such cases.
- These methods cannot detect multiple faces since the required features would mislead the rules.

2.5 Template Matching Techniques

Matching is widely applicable approach to locate objects in images. While matching is often based on gray-level properties of image sub-regions, it can be well performed using image-derived features or higher-level image descriptors (Sonka *et al.*, 2008). There are many ways to develop a standard face template as a whole or the facial features separately. These techniques can be performed in either the spatial domain or frequancey domain. The correlations between

an input image and the predefined templates are computed first to detect candidate locations of faces. Then, matching with the other sub-templates is performed at the candidate positions.

Although this approach is simple to implement, it is inadquate to deal with variation in face appearance such as scale, pose, and shape. Therefore, multiresolution, subtemplates, and deformable templates have subsequentlty been proposed to achieve scale and shape invariance (Yang *et al.*, 2002). Templates can be divided into two categouries: predefined templates and deformable templates.

2.5.1 Predefined Templates

A hierarchical template matching method for face detection was proposed by Miao *et al.* (1999). The system consists of four stages: the first stage is a preprocessing step; the next three stages comprise the gravity-center template matching, gray-level check, and edge-level check. The system transforms the source image into a mosaic image with different scales. When the scale of the mosaic unit is proper, front face components will be simultaneously contained in four rows of the mosaicized image. In a mosaic image, if human faces exist, there will be some pieces of serial units whose gray values are smaller to those of the units around them. Some of these pieces possibly correspond to the areas of eyes, eyebrows, nose bottom and mouth. Laplace operator is used to extract horizontal edges in the mosaic image, and then the positions of the gravity centers of the candidate edges are calculated. Then the system searches for faces in a skip mode, that is, to scan only the gravity centers where faces probably exist rather than searches the full mosaic image. Four gravity center templates are designed for matching the points in scanned areas in the gravity center picture. The system uses tens of matching rules through plenty of experiments. The test dataset consists of 290 images collected from image library and TV, which are made up of four categories; these are: single front face, single rotated face, single face with glasses, and multi-face images. The reported detection rates using these test images are 93.8%, 88%, 64.7%, and 70.2% respectively. The reported average detection rate is 83.8% with false detection rate of 2.5%.

Cai and Goshtasby (1999) used front and side-view face models to verify the presence of a face in a candidate skin color region. The frontal model was found by averaging intensities of 16 front-view faces of men and women without facial hair or glasses. The data set images were selected from Yale University and University of Bern face databases. These face models were used in a template-matching process to detect faces within skin regions. The authors reported very low FN/FP error rate of about four percent for an image size 300 × 220 pixels containing four or fewer faces (Cai & Goshtasby, 1999).

Chen *et al.* (2009) proposed a half-face template for face detection problem. The idea is that full face-template may be regarded as a combination of the almost symmetrical left face template and the right one. The system computes the similarity between the templates and the faces of different angles in images. The value of half face template lies in the fact that it can reduce about half of the time complexity in face detection so that the detection speed can be increased. The face detection is carried out using the method of template matching to determine the position of the face in the image. The experimental results indicate that the half face-template can adapt the side face images in a large angle, which improves face detection at a different pose. This method achieves 82.5% detection accuracy on a database of images containing 120 faces.

Ghazali, Ma, and Xiao (2012) proposed a face detection method consisting of two stages. First, skin color information is used to estimate face candidate regions. In the second stage, template matching approach is used to compute the cross-correlation value. The threshold value for classifying a segment as a face area is determined empirically. The reported detection precesion rate is 95.45% with a false positives rate of 4.62% using 100 test images collected from the internet.

2.5.2 Deformable Templates

One of the earlier approaches to detect and describe facial features was proposed by Yuille, Hallinan, and Cohen (1992) for purposes of face recognition. The deformable templates are specified by a set of parameters which enables a priori knowledge about the expected shape of the features to guide the detection process. The templates are flexible enough to be able to change their size, and other parameter values, so as to match themselves to the data. The final values of these parameters can be used to describe the features. An energy function is used to link features such as edges, peaks, and valleys and then interacts dynamically by altering their paramters to minimize the enrgy function.

Kwon and Lobo (1994) proposed a face loclization approach based on snakes. The approach comprises two stages: finding candidate faces based on snakes and then confirming the existance of the facial features. In the first stage, mutliple snakelets curves are dropped on the input image where they become algined to the natural wrinkls and curves of facial image. When these snakelets have stabilized, those snakelets that have found shallow valleys are deleted. In the second stage, seven steps are used to confirm the existance of the face; these are, finding the initial rough face oval, chin, face sides, eyes, mouth, and finding nose. Rules concerning the ratios of distances between facial features are used and compared with the previousely stored reference rations. If a substantial number of the facial features are passed these rules, the presence of a face is considered to be confirmed. No quantitative results are provided.

2.6 Illumination Variation Problem

Faces and other parts of the human body recorded under natural environments are frequently subject to illumination variations which affect their color appearance (B. Martinkauppi, Soriano, & Pietikainen, 2003). One of the significant features in the human vision system is the ability to automatically adapt itself to detect colors by disregarding the effects of varying illumination. This ability aids in keeping the object's color appearance stable (i.e. relatively constant) and it is

known as "*color constancy*" (Ebner, Tischler, & Albert, 2007). For example, a green apple looks green to us at midday sunlight illumination, and also at sunset.

Unfortunately, computer vision systems (i.e. including digital color cameras) do not have this kind of "built-in" mechanism in order to adapt themselves to color changes caused by illumination variations. Therefore, the same image scene can be greatly different under different lighting conditions. Furthermore, different color cameras do not necessarily produce the same color appearances for the same scene under the same imaging conditions (J. B. Martinkauppi & Pietikäinen, 2005).

In general, due to difficulty in controlling the lighting conditions in practical applications, variable illumination is one of the most challenging problems associated with face detection/recognition (Xie & Lam, 2005). For instance, let us consider the facial images shown in Figure 2.19. Figure 2.19(a) shows a set of face images captured under different lighting conditions for the same individual. Figure 2.19(b) shows a set of face images captured under similar lighting condition but for different individuals. As shown in this figure, it is clear that illumination variation causes dramatic changes in the face appearance. Many works have found that variations between images of the same face due to illumination are almost always larger than image variations due to change in face identity (Adini, Moses, & Ullman, 1997; An, Wu, & Ruan, 2010; Xie & Lam, 2005).

Figure 2.19: Image variations due to illumination; (a) image variations due to illumination for the same face; (b) image variations due to change in face identity.

Due to the fact that the intensity values of the pixels in the face image are usually affected by lighting conditions, most of the face detection/recognition methods could not achieve good performance on the outdoor face databases (An *et al.*, 2010). Varying illumination and shadows cast by facial features may cause many sub-problems such as:

- False color tone.
- Loss of feature information.
- Shape distortion of objects.
- Failure of conjugate image matching within the shadow area.

Therefore, image enhancement and normalization pre-processing becomes a significant part of most face detection/recognition systems (M. Lee & Park, 2008). Xie and Lam (2005) stated that the recognition rate can be improved by 53.3% to 62.6% when the algorithm for lighting compensation is used. Practically, even when applying histogram equalization to images with controlled illumination, the performance of face detection/recognition is substantially improved (Zou, Kittler, & Messer, 2007a).

Recently, a number of approaches have been proposed to solve the lighting problem (An *et al.*, 2010). These approaches can be classified into four main categories, namely 1) image enhancement and normalization pre-processing, 2) color constancy, 3) illumination invariant features, and 4) face modeling under varying illumination.

2.6.1 Image Enhancement and Normalization Pre-Processing

These approaches aim at remedying the illumination variation problems by employing imaging techniques to improve the quality of image, normalize illumination, and correct colors. Some approaches are global, in the sense that pixels are modified by transformation function of an entire image. Although the global approach is suitable for the overall image enhancement, there are cases in which it is necessary to enhance details over small areas in an image (i.e. local

enhancement). The Histogram Equalization (HE) method is mostly used in image enhancement. Illumination variations may result in an image whose histogram covers only a portion of the available brightness values for display. Expanding the brightness scale by spreading the histogram to full range may improve the visibility of features (Russ, 2007). Xie and Lam (2005) proposed illumination compensation algorithm, namely, Block-based Histogram Equalization (BHE) to estimate the category of light source in the original face image. Shan *et al.* (2003) proposed Gamma Intensity Correction (GIC) method to normalize the overall image intensity at the given illumination level. The region-based strategy, combining GIC and the (HE) is used to further eliminate the side-lighting effect. Du and Ward (2005) presented a wavelet-based normalization method to enhance the contrast as well as the edges of face images. Recently, An *et al.* (2010) proposed an illumination normalization model (INM) combined with HE for the pre-processing of face recognition under varied lighting conditions.

The main drawback of most image enhancement approaches is that they are inadequate to be generalized for all kind of images. Global transformation such as histogram equalization-based methods cannot correctly improve all parts of the image. When the original image is irregularly illuminated, some details on the resulting image will become either too bright or too dark. Figure 2.20(a), adopted by Starovoitov *et al.* (2003), depicts a face image with non-uniform lighting. Its histogram is shown in Figure 2.20(b). Histogram equalization-based approach is used to enhance the image and the result is shown in Figure 2.20(c). In this figure, although the left side of the face image is properly enhanced and some facial features became more visible, the right side of the face image becomes too bright so that some facial features are lost.

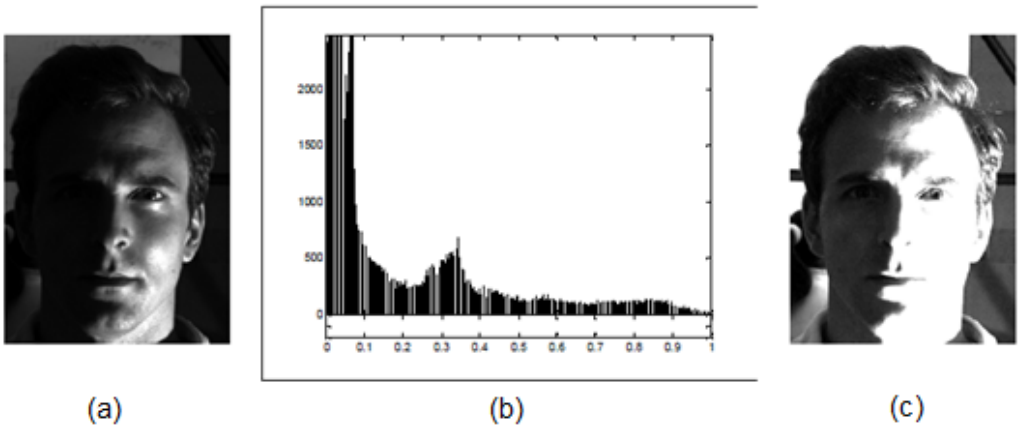


Figure 2.20: Image enhancement using histogram equalization-based approach image; (a) source image; (b) its histogram; (c) newly-generated image, adopted from Starovoitov, *et al.* (2003).

The problem of global image enhancement turns the attention of many researchers to the local enhancement. It is necessary to enhance details over small areas in an image, while leaving the other areas untouched. The main drawback of many existing local lighting correction approaches is the high computational cost at run time. Since human face may appear at any place in an input image, the system applies a sliding window over the entire image searching for faces. The system should carry out the lighting correction step on each sliding window which causes a serious delay.

2.6.2 Color Constancy

Color constancy is inspired by human vision, which is a well-known capability of the human vision system to discriminate colors under varying illuminants. Computational color constancy algorithms aim to correct color deviations caused by a difference in illumination as done by the human vision system. A number of algorithms have been developed to compute color constant descriptors of objects in view, irrespective of the light which illuminates the scene. The descriptors can be used afterwards to correct colors. These include several Retinex algorithms that attempt to estimate the reflectance of each point (Morel, Petro, & Sbert, 2009). Barnard, Finlayson, and Funt (1997) presented an algorithm which uses information from both surface reflectance and illumination variation for color constancy. Provenzi *et al.* (2007) proposed an automatic color equalization (ACE) based on a computational model inspired by some

adaptation mechanisms of the human vision. (Ebner, 2006) used genetic programming to evolve an algorithm for color constancy. Agarwal (2006) stated that most algorithms make some assumption about the statistics of the reflectance to be encountered, and most make assumptions about the illuminants that will be encountered. Moreover, most machine color constancy approaches cannot handle situations with more than one illumination present in the scene. More details on color constancy algorithms also can be found in the reported work by (Agarwal, Abidi, Koschan, & Abidi, 2006; Ebner, 2007).

2.6.3 Illumination Invariant Features

These methods are based on representations of a face image that are relatively insensitive to changes in illumination. These representations are directly used for face detection and recognition. They include edge map (Ghiass & Fatemizadeh, 2008), image intensity derivatives (C. H. T. Yang, Lai, & Chang, 2002), and image convolved with a 2D Gabor-like filter (Adini *et al.*, 1997). But An *et al.* (2010) stated that "The illumination invariant features are limited and not enough for recognition in large scale face databases. Under the largely varied lighting conditions, these methods achieve low level performance". A good survey on illumination invariant face recognition can be found in the work by (Zou et al., 2007a).

2.6.4 Face Modeling under Varying Illumination

These methods are based on a statistical or physical model. For statistical modeling, no assumption concerning surface property is needed (Zou et al., 2007a). Hallinan (1994) argued that five eigenfaces are sufficient to represent the face images under non-uniform lighting condition. Wang *et al.* (2007) also presented generalized image-pre-processing PCA-based face recognition algorithm.

In physical modeling, a 3-D face model is used to synthesize the virtual face image from shading based on the assumption of surface reflectance properties. The *Photometric Stereo* is a technique to estimate local surface orientation by using several images of the same surface taken

from the same viewpoint but under illumination from different directions. Ping-Sing and Shah (1994) proposed a model-based light source identification method to improve existing *Source-From-Shading* (SFS) algorithm. Zhao and Chellappa (2000) presented a method based on *Symmetric Shape from Shading* for illumination of insensitive face recognition. Shashua and Riklin-Raviv (2001) and Wang , Li, and Wang (2004) proposed a *Quotient Image* (QI) which treats face as an Ideal Class of Object (ICO).

2.7 Summary

Up to date literature review for face detection methods is presented in this chapter. Based on this literture review we found that most face detection systems imply some kind of rule-based approach to guide the system. Some methods imply preconditions and assumptions to make the detection easier and faster. Many methods were applied successfully for face localization although the authors used the term "face detection". Most of the face detection methods focus on detecting frontal faces with uniform lighting conditions.

Note that the general performance of a detection system depends on several factors such as FP rate, FN rate, true (or positive) detections rate TP, detection time, learning time, the amount of training samples required, algorithmic parameters, benchmarking data sets, etc. A fair evaluation should take all these factors into consideration.

Although different methods use different face databases in training and testing, this chapter shows an overall picture about the performance of the methods. In general, each method has its advantages and disadvantages. A comparative analysis of our face detection system with other systems is presented in Chapter 7.

In the last two decades, a significant progress has been made but there is still work to be done. However, fully automatic face detection from complex images is still a challenging and interesting problem.

Varying illumination and shadows cast by facial features may cause false color tone, loss of feature information, shape distortion of objects and failure of conjugate image matching within the shadow area. Although most of the images that we want to deal with are 2D color images, until recently many face detection and recognition approaches such as PCA, Neural Networks, Wavelet Transforms, HMM, SVM, etc. are done using gray scale images. These approaches are very sensitive to gray values of the face image.

Generally speaking, it is preferable to apply image enhancement before any pattern analysis. This will adjust color features between the different illuminations and consequently improve the detection rate. In general, the best transformations for modifying image illumination normally are selected interactively. The idea is to adjust experimentally the image brightness and contrast to provide maximum detail over a suitable range of intensities. Unlike the interactive enhancement, the transformations can be applied to color images in an automated way.

In this thesis, devising a new method for automatic illumination enhancement in application to the face detection problem is considered one of the research goals.

CHAPTER THREE

SKIN COLOR MODELING AND DETECTION – BACKGROUND

3.1 Introduction

In digital images, when we want to analyze, interpret, or understand an image automatically, we have to identify unambiguously the pixels that belong to a particular object of interest. The process of such identification is known as *segmentation* (Efford, 2000). Sonka *et al.* (1999) defined image segmentation as follows: "Its main goal is to divide an image into parts that have a strong correlation with objects or areas of the real world contained in an image". According to Gonzalez *et al.* (2002) segmentation subdivides an image into its constituents or objects. Cheng *et al.* (2001) defined image segmentation as "the process of dividing an image into different regions such that each region is, but the union of any two adjacent regions is not, homogeneous". Skarbek, Koschan, & Veroffentlichung (1994) opted for a simpler interpretation, that is "the identification of homogeneous regions". Fu & Mui (1981) defined image segmentation as "the division of an image into different regions, whereby each region has certain properties". From our point of view, the aim of segmentation is to group together pixels that have similar properties, according to a set of predefined criteria, and as a result divides the source image into a set of regions or objects. The regions of pixels that we generate should be meaningful. The level to which the subdivision is carried depends on the problem being treated.

In most systems, the segmentation only subdivides an image; it does not attempt to recognize the individual segments or their relationships to one another. Generally an individual pixel cannot indicate whether it is located in a region. Hence the neighborhood of the pixel needs to be analyzed. If the pixel shows the same color value (or other properties) as its neighbors, then we have a good reason to believe that it lies within a region of constant values. Image segmentation (gray or colored) is an important pre-requisite for many computer vision and information retrieval systems. Segmentation is generally the first step in image analysis, object

detection, and pattern recognition (Gonzalez *et al.*, 2002). Therefore, we can say that segmentation techniques can be found in many applications involving the detection, recognition and measurement of objects in images.

Sonka *et al.* (1999) divided image segmentation task into two types (complete and partial image segmentation):

- **Complete image segmentation** results in a set of disjoint regions corresponding uniquely with objects in the source image. Hence, it can be formulated as follows: Let R represent the entire image region. Segmentation can be viewed as a process that partitions R into *n* sub-regions $R_1, R_2, R_3, ...R_n$, such that:

$$\bigcup_{i=1}^{n} R_i = R \quad and \quad R_i \cap R_j = \emptyset \ for\ all\ i\ and\ j, \qquad where\ i \neq j$$

  The first condition implies that every image pixel must be in a region. This means that the segmentation algorithm should not terminate until every pixel is processed. The second condition means that no pixel can belong to more than one region.

- **Partial image segmentation**: segmentation should stop when the regions of interest are isolated. A reasonable aim of many image processing and vision applications is to apply partial segmentation where an image is divided into separate regions with respect to a chosen property such as brightness, color information, texture information, reflectivity, etc. This would reduce the size of data to be processed. The substantial reduction in data volume offers an immediate gain, which is important to most applications.

From the point of segmentation bases, Gonzalez *et al.* (2002) classified image segmentation algorithms into two basic categories: discontinuity and similarity. In the first category, the approach is to partition an image based on abrupt changes in intensity, such as point detection, edge detection, and boundary detection. The principal approaches in the second category are based on partitioning an image into regions that are similar according to a set of predefined

criteria. Pixel-based segmentation, region-based segmentation, region splitting and merging, and watershed segmentation algorithms are examples of methods in this category.

Segmentation algorithms can be done interactively or automatically. Suitable and excellent segmentation methodologies are usually interactive, that is, under control of users, and partition the image into non-overlapping regions of interest. Automatic image segmentation algorithms, designed to partition an image into regions, might be required for most of the vision-based systems and it would be ideal for automatic object detection. Efford N. (2000) stated that a reliable and accurate image segmentation is generally very difficult to achieve by purely automatic means.

Since image segmentation is the first step in image analysis, the accuracy of segmentation determines the eventual success or failure of computerized image analysis procedures. For this reason, considerable care should be taken to improve segmentation accuracy.

The subsequent sections below discuss skin color information, properties of human skin, challenges of skin color detection, image segmentation based on skin color, color spaces, skin color modeling and region-based approaches.

3.2 Why Skin Color Information

Until recently many of the face detection methods such as PCA, ANN, SVM, HMM, etc., were done at the intensity level using gray scale images. The segmentation of gray images is a very difficult task as in general, intensity information does not provide enough information as color images. Therefore, most of the these methods are based on appearance-based approaches that imply high computational cost as well as high false detections even just for the task of detecting a single face in an image. Here are the main problems (Jin et al., 2007):

- Sliding-window technique is widely used to search for faces but it is very time consuming as the source image is treated without any analysis of its contents (i.e. only low-level processing). The sliding window has to be applied at every pixel location in source image, which makes the search space relatively high. Among the millions of sub-images generated from an input image, only very few contain faces. The occurrence of a face in an image is a rare event (J. Wu, Brubaker, Mullin, & Rehg, 2008). In this sense, face detection and all other detection problems in vision are rare event detection problems. As the number of scanning windows in these methods is high, the probability of false positives FPs is higher than other methods.
- An appearance-based method is scale sensitive. To detect faces at different scales, the source images have to be resized several times (sub-sampling) and search is repeated.
- To detect faces with varying orientations, input images have to be rotated several times.

Therefore, it is needed to process local information in a very short period of time in order to identify "hot spots" (or “regions of interest”) which are likely, though not certain, to contain a desired object or class of objects (i.e. human face). Then, more complex classifier that usually requires intensive high processing are used to make the final arbitration of whether these "hot spots" correspond to objects of interest or not.

The use of color as a valuable feature in image analysis, and specifically in image segmentation, is motivated by the following principal factors:

- Color images can provide more information than gray level images. Often, when the objects cannot be extracted using gray scale, they can be extracted using color information. For example, two objects of similar gray tones might be very different in color. Hence color is a powerful feature that often simplifies object detection and extraction from a scene (Gonzalez *et al.,* 2007). Since faces could be anywhere in an arbitrary image, traditional face detectors usually have to examine every pixel in the image. With the goal to detect faces from an image in a faster and more accurate manner, a good strategy to reduce the search space for human targets is required.

Human skin segmentation (i.e. Color-based image segmentation) can restrict the search region. We will show in the subsequent chapters that most of the colors in the color space set are non-skin color (i.e. about 90.36%). So, as our strategy is to reduce the search space for human targets, the image is initially segmented into "skin" and "non-skin" regions based on pixels' color. This is important to speed up the face detector system because it will avoid the exhaustive search for faces at background.

- Color is robust against object rotation, scaling, and partial occlusion.
- The processing of color information has proven to be much faster than processing of some other features.
- The background in complex images usually contains objects and patterns that look like face patterns. An advantage of skin color segmentation is to avoid the exhaustive search at background (i.e. the background will be excluded). As the number of scanning windows is lower than before, the probability of false detections is less which improves the accuracy of the system.
- The advances in computers are increasing rapidly, and PCs can be used to process color images now with high speed processors and low cost storage devices. This creates a situation that most of images we really want to deal with are 2D colored images (i.e. in contrast to 2D grayscale or 3D images).
- Color information can be used as complementary information to other features to improve detection rate.

For the above-mentioned reasons, human skin detection becomes an important step for many human-related image processing applications such as face detection/recognition, video surveillance systems (Gejguš & Šperka, 2003; Habili, Lim, & Moini, 2004; Kim, Park, & Joo, 2005; Zaqout, 2006), naked image filters (Chin, 2008; Duan, Cui, Gao, & Zhang, 2002; J. S. Lee et al., 2007; Rowley, Jing, & Baluja, 2006; Sevimli et al., 2010), and hand gesture recognition (Ruijsscher, 2006; M. H. Yang, 2000).

3.3 Properties of Human Skin

Body organs are not all internal like the kidney or heart. Skin which appears on the outside is our largest organ (Tobin, 2006). Skin color appearance gives us a sign about the person's race, mode, healthy, and the age. Human skin consists of three main layers (epidermis, dermis and subcutaneous) as shown in Figure 3.1 (Martinkauppi 2002). Each has its function and all skin layers work together.

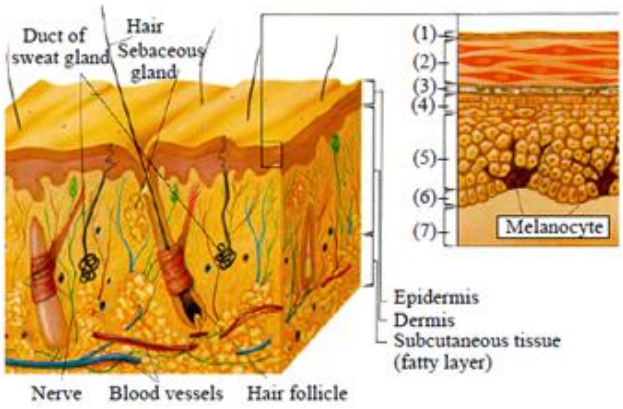


Figure 3.1: The structure of the human skin. (1) Keratin; (2) Horny layer; (3) Lucid Layer; (4) Granular Layer; (5) Spinous layer; (6) Basal layer; (7) Dermis; Adopted from (Martinkauppi 2002)

The three main skin layers are:

- Epidermis layer is made of skin cells at the end of their life-cycle. These cells provide protection from injury and a barrier for infectious organisms. The epidermis holds in fluid and protects raw nerve cells from too much stimulation.
- Dermis layer contains capillaries that feed the cells with nutrient-rich blood. Most things grow here - including hair follicles, nerve cells, and sweat glands. If damaged, the dermis will weep serous fluid and swell.
- Subcutaneous layer (also known as the hypodermis) is technically not officially skin, but rather attaches the skin to everything beneath. It also contains a layer of fat. Some of us have more fat than others, but this layer is always present in some form. This layer allows for drainage. Blood vessels in the subcutaneous layer feed and drain the capillaries of the dermis.

The outer surface of skin is covered with dead cells causing no regular reflection, while the glossiness of skin can be because of sweat or skin oil. Complex phenomena can happen during the interactions between incident light and skin (Storring, 2004). The final skin spectra are formed by the interaction between skin and light: light striking skin is transmitted, absorbed, and reflected through the layers. The spectra for human skin generally form a continuous homologous series because of characterization caused by absorption of *melanin* and *hemoglobin*, in which melanin in epidermis and hemoglobin in dermis play dominant roles and mostly decide the skin appearance (HU, 2011).

3.4 Challenges of Skin Color Modeling and Detection

The principal idea of most existing skin color models is based on the assumption that skin color is quite different from colors of other objects and its distribution forms a cluster in some specific color space. From our experience in this work we found that this is true when dealing with a constrained environment. Detecting skin-colored pixels, although seems a straightforward easy task, has proven to be quite a challenging task in complex images that are captured under unconstrained imaging conditions (Kakumanu *et al.*, 2007).

Working with skin color as a cue feature for face detection is influenced by the following challenging factors:

- **Illumination variations**: when dealing with colors, the illumination variation is the most important problem that seriously degrades the segmentation performance (Storring, 2004). A change in the light source distribution or in the illumination level (indoor, outdoor, highlights, shadows, non-white lights) produces a change in the color of the skin. Usually, the dark shadow on the face is a result of strong directional lighting that has partially blackened some facial regions. This is due to the non-plane shape of the facial features. Sometimes, a face has a “bright spot” due to reflection of strong lighting.

- **Different ethnic groups (Race):** skin color appearance varies from person to person due to physical differences among human racial groups. For example, Europeans (Caucasians), Africans, Asians, etc. have different skin colors that range from white, to brown to dark (Tan *et al.*, 2012).
- **Imaging conditions:** When the image is formed, factors such as camera characteristics (sensor response, lenses) affect the skin appearance. In general, different color cameras do not necessarily produce the same color appearances for the same scene under the same imaging conditions (Yang *et al.*, 2002).
- **Image montage and reproduction:** different image collections from the Internet, movies, newspapers, and scanned images are usually uncontrollable and have virtually unlimited sort of montage processes. There are tools to reproduce skin tones including setting new pigment concentration and changing the color of skin image by applying color transfer technology. Some images already have been captured with the use of color filters. This makes dealing with color information even more difficult.
- **Makeup:** affects the appearance of skin color (Kakumanu *et al.*, 2007).
- **Complex background:** is another challenge that comes from the fact that many objects in the real world might have skin-like color. For example, furniture, clothes, blond hair, rocks, etc. The diversity of backgrounds is virtually unlimited. This causes the skin detector to produce false detections in the background when the environment is unconstrained (Kakumanu *et al.*, 2007).
- **Aging:** Human skin is very fresh and elastic in one's young days. The tension of skin is lost and becomes dry as one grows older. The changes according to aging have a series of transitions from fresh skin to dry skin and then dry rough skin with wrinkles (Storring, 2004).

With numerous images types and sources, skin color can vary dramatically in its appearance, making accurate image segmentation based on skin-color very difficult task.

On the other hand, false-colors image is another problem that degrades image segmentation. For example, blue sky in a normal scene might be converted to appear red, and green grass may be transformed to blue. There are three possible reasons for mapping real normal color to false-color images (Pratt, 2001): ***1)*** to place normal objects in a strange color world so that a human observer will pay more attention to objects than if they were colored normally; ***2)*** the attempt to color a normal scene to match the color sensitivity of the human viewer; and ***3)*** to produce a natural color representation of a set of multispectral images of a scene. Some of the multispectral images may even be obtained from sensors whose wavelength response is outside the visible wavelength range, for example, infrared or ultraviolet.

In this work, we consider this issue as being far from our direction and goal of skin segmentation for feasible system and solutions of research focus, significant research findings, and system efficiency. The false-color problem can be treated as a preprocessing step. Even images containing strong colors that tend to be unreal colors are discarded, such as images that completely tend toward blue, violet, etc.

3.5 Image Segmentation Based on Skin Color

In recent years, there has been an increasing interest in the problem of human skin segmentation under unconstrained imaging conditions (i.e. complex color images) (Storring, 2004). An improved human skin segmentation approach with higher detection rate enhances the performance of the many computer vision applications based on the skin detection operation.

As mentioned before, skin detection aims to segment the input image into regions in order to locate the potential face areas based on the color feature (i.e. usually called “regions of interest” or “skin-maps”), and consequently excluding background regions. The high importance of this stage is based on quite clear fact: image segmentation is the first step in image analysis and if a certain skin region is missed (e.g. face region is misclassified), it cannot be retrieved by the subsequent stages of the system.

Skin detectors can be very fast. Thus, many face detectors often use skin detectors as a pre-requisite to higher level image processing stages (H. Y. Chen et al., 2008; Garcia & Tziritas, 1999; Moallem et al., 2011; Zaqout et al., 2004). This step becomes more important and essential when dealing with high resolution images (because the search space will be larger as the resolution of image becomes higher). When we want to build a skin detector, three main questions are formulated (Vezhnevets, Sazonov, & Andreeva, 2003):

i) **What color space to choose?**

This implies understanding color fundamentals. How are the colors represented in digital images? What do we mean by color spaces? The difference between different color spaces is an essential issue to perform many image processing tasks such as manipulating colors, segmentation, color correction, etc. The choice of the color space affects the shape of the skin-color cluster, which affects the detection process.

ii) **How exactly should skin color distribution be modeled?**

Most existing skin segmentation approaches assume that human skin color is quite different from colors of other objects and its distribution forms a cluster in the color space. Skin samples (or skin patches), usually called or *training data*, to build skin-color clustering model are needed, followed by a way to build a representative model that describes this clusters to a certain degree of accuracy and correctness. Since the final goal is to classify an unknown new pixel or region as having a color like skin color or not, it is an important issue to choose the most suitable classification algorithm.

iii) **How the actual segmentation is done?** For example, pixel-based or region-based segmentation (i.e. by taking the spatial relationship into consideration as well).

3.6 Color and Color Spaces

Color is not an intrinsic property of an object itself. It is the perception of the energy emitted or reflected from it. When light hits objects, some wavelengths are absorbed and some are reflected, depending on the object materials. The reflected wavelengths are perceived as the object's ***color***. Visible light is made of seven wavelength groups: red, orange, yellow, green, blue, indigo, and violet as shown in Figure 3.2, adopted from (Gonzalez & Woods, 2002). The reddish, greenish and violet colors are respectively the long, mid-size and short wavelengths.

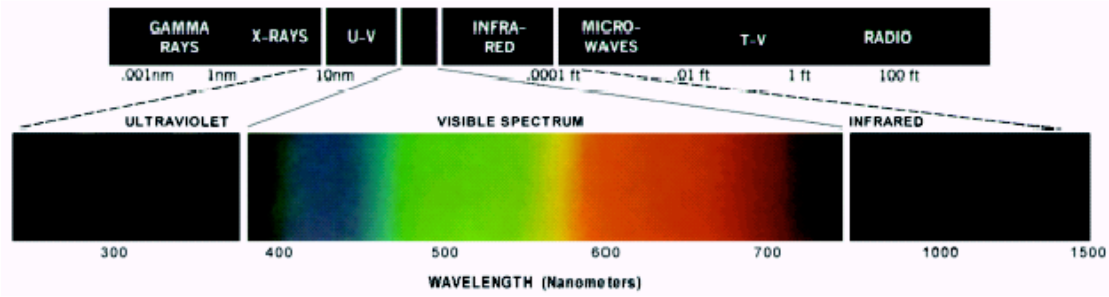


Figure 3.2: Wavelengths comprising the visible range of electromagnetic spectrum, adopted from (Gonzalez & Woods, 2002).

At its core, digital imaging works by transforming colors into numbers either by using the physics of light waves, or the way the eye perceives color, or the way ink creates colors. Each of these methods is useful in different ways.

Color space is a mathematical model to facilitate the specification of colors in some standard way to denote, create and visualize colors which can be represented as *tuples* of numbers, typically as three or four values of color components or channels. The color space represents a coordinate system where each specific color is represented by a single point in the coordinate system. As for any mathematical representations of physical phenomena, color models can be expressed in different ways. Figure 3.3 shows an example of how the colors are represented as numbers using a well known RGB color space, adopted from Image Processing Toolbox (IPT) MATLAB (2010).

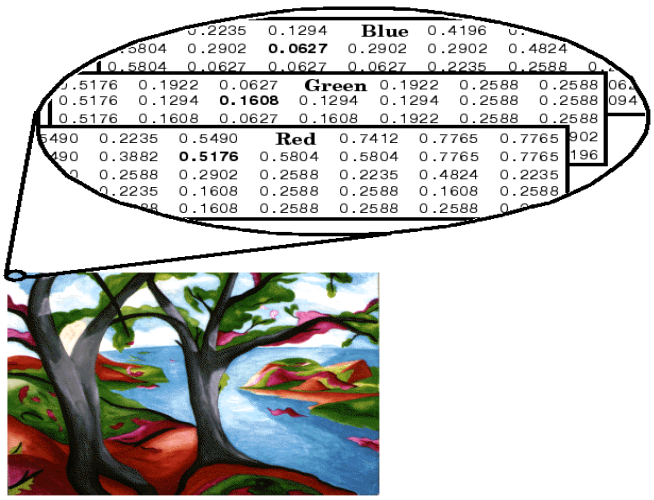


Figure 3.3: Representation of colors in digital images as numbers using RGB color space, adopted from (MATLAB 2010).

Each model is oriented toward some hardware or application needs. The various color spaces exist because they present color information in ways that make certain calculations more convenient or because they provide a way to identify colors that is more intuitive (Russ, 2007; Sonka *et al.*, 2008). The most widely used are classified as (Kakumanu et al., 2007; Moallem et al., 2011):

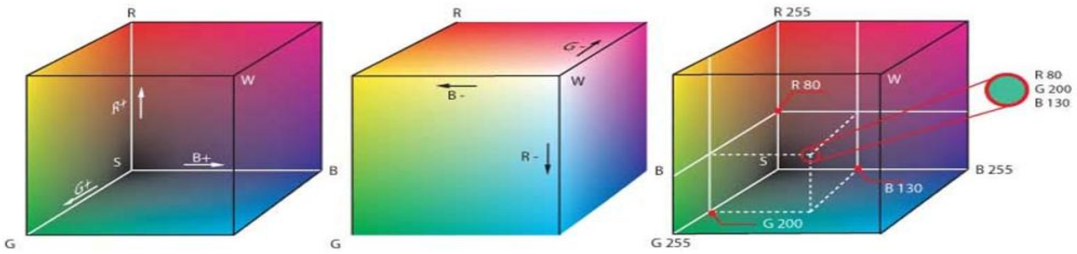
- **Basic color spaces** (RGB, Normalized RGB) (Dargham & Chekima, 2006; Eveno, Caplier, & Coulon, 2001; Kovac, Peer, & Solina, 2003; Zaqout *et al.*, 2004);
- **Perceptual color spaces** (HSI, HSV, HSL, TSL) (Fernandez, Ortega, Cancela, & Penedo, 2012; Garcia & Tziritas, 1999; Moallem et al., 2011; Terrillon, David, & Akamatsu, 1998; Tomaz, Candeias, & Shahbazkia, 2004).
- **Orthogonal color spaces** (YCbCr, YIQ, YUV, YES) (Albiol et al., 2000; Hsu, Abdel-Mottaleb, & Jain, 2002; Z. Liu, Yang, & Peng, 2005; Vadakkepat, Lim, De Silva, Jing, & Ling, 2008).
- **Perceptually uniform color spaces** (CIE-Lab, CIE-XYZ and CIE-Luv) (Cai & Goshtasby, 1999).
- **Others** - Mixture spaces, color ratio spaces (Gomez, Sanchez, & Sucar, 2002; Raghuvanshi & Agrawal, 2012; Wang & Yuan, 2001; Yuetao & Nana, 2011).

To cope with the skin detection challenges, many groups have centered their study on selecting the color space most suitable for skin detection (Chaves-González, Vega-Rodríguez, Gómez-Pulido, & Sánchez-Pérez, 2010; Kakumanu *et al.*, 2007; Vezhnevets *et al.*, 2003) . Many users, digital cameras, color monitors, and other storage devices employ the RGB model as the default color space to display and store digital image data. However, some applications may find it more convenient to use other color spaces.

The remainder of this section is focused on the most widely used color spaces along with a comparison.

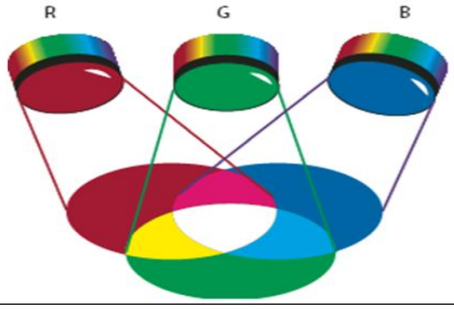
3.6.1 RGB Model & Normalized RGB

RGB model is based on the Cartesian coordinate, where each color is represented in its three primary components: Red (R), Green (G), and Blue (B). It can be geometrically represented in a three-dimensional cube with three axes perpendicular to each other as in Figure 3.4(a). The primary colors R, G and B are at the corners (255,0,0), (0,255,0), and (0,0,255); black is at origin (0,0,0) and white at the opposite corner (255,255,255). The RGB model uses the fact that: a wide variety of colors can be obtained by mixing red, green, and blue light in different proportions as in Figure 3.4(b). The different colors in the RGB model are points on or inside the cube.



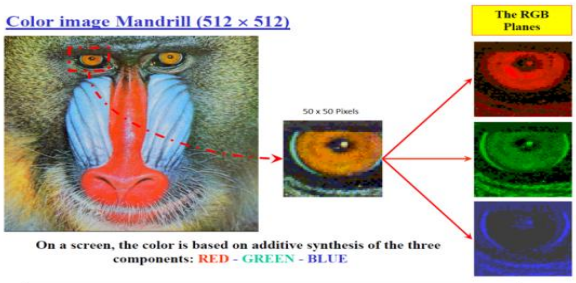
(a)

The RGB color cube.



(b)

Generating colors in RGB color model.



(c)

The 3 planes of the RGB image.

Figure 3.4: The RGB color Space; (a) RGB cube; (b) Generating colors in RGB model; (c) The 3 planes of the RGB image.

Since the RGB model is the most commonly used, we have to enlighten the main properties of this model (Gonzalez & Woods, 2002; Ma & Leijon, 2010; Russ, 2007; Sonka *et al.*, 2008):

- Each point (color) within the cube is specified by three numbers (R, G, B triple).
- The diagonal line of the cube is from black (0,0,0) to white (255,255,255), representing all the gray levels, in which, the three components R, G, and B are the same.

- In general, different computer hardware/software combinations will use different ranges for the colors.
- The RGB color space is an "*additive*" model as in nontechnical terms, its origin starts at black, and all other colors are derived by adding intensity values.
- The RGB cube is smaller than our visible range and represents fewer colors than what we can see (H. E. Burdick, 1997).
- The RGB colored image of screen dimension M rows and N columns is an M × N × 3 of colored pixels. Here, 3 represents the three layers of Red, Green, and Blue intensities, see Figure 3.4(c).
- RGB is suitable for technical applications, but it is of limited use for image segmentation and analysis because of the high correlation between the R, G, and B components. It is not a perceptual model (Efford, 2000); that is, the R, G, and B components contain both the color and illumination (or darkness) information.
- The main advantage of the RGB space is its simplicity.

The RGB color model is implemented in different ways, depending on the capabilities of the system used. The most common system used is an 8-bit per one component to describe a color, resulting in the 24-bit implementation (i.e. the number of bits required to represent a pixel in a colored image). Any color space based on such a 24-bit RGB model is thus limited to a range of 256×256×256 ≈ 16.7 million colors (Ma & Leijon, 2010). Some implementations use more bits per component (e.g. 16 bits), resulting in a larger number of distinct colors. Normalized RGB is a simpler representation of RGB by a simple procedure:

$$\begin{aligned} R_n &= R / (R+G+B); \\ G_n &= G / (R+G+B); \\ B_n &= B / (R+G+B) \end{aligned} \tag{3.1}$$

The third component B_n can be omitted because it does not hold any significant information (i.e. It is known that $R_n + G_n + B_n = 1$). Thus, reducing the space dimensionality is preferable.

The RGB and normalized RGB color space was used for skin segmentation by (Brand & Mason, 2000; Chen & Wang, 2007; Dargham & Chekima, 2006; Eveno *et al.*, 2001; Frisch *et al.*, 2007; Jones & Rehg, 2002; Kovac *et al.*, 2003; Skarbek *et al.*, 1994; Soriano, Martinkauppi, Huovinen, & Laaksonen, 2000; Taqa & Jalab, 2010; M. H. Yang & Ahuja, 1998; Zaqout *et al.*, 2004).

3.6.2 CMY and CMYK models

The CMY model is based on the secondary colors (Cyan, Magenta, and Yellow) and it is used for the offset printing process that is based on pigments (inks). It is called "*subtractive"* model and is opposite to an additive color model. In additive color (e.g. RGB), white is the additive combination of all primary colored lights, while black is the absence of light. In the CMY model, it is the opposite: white is the natural color of the paper or background, while the mixture of all three colors will produce other colors including black as in Figure 3.5(a-b). The subtractive color theory underlying CMY, holds that various levels of cyan, magenta, and yellow absorb or "subtract" a portion of the spectrum of the white light illuminating an object.

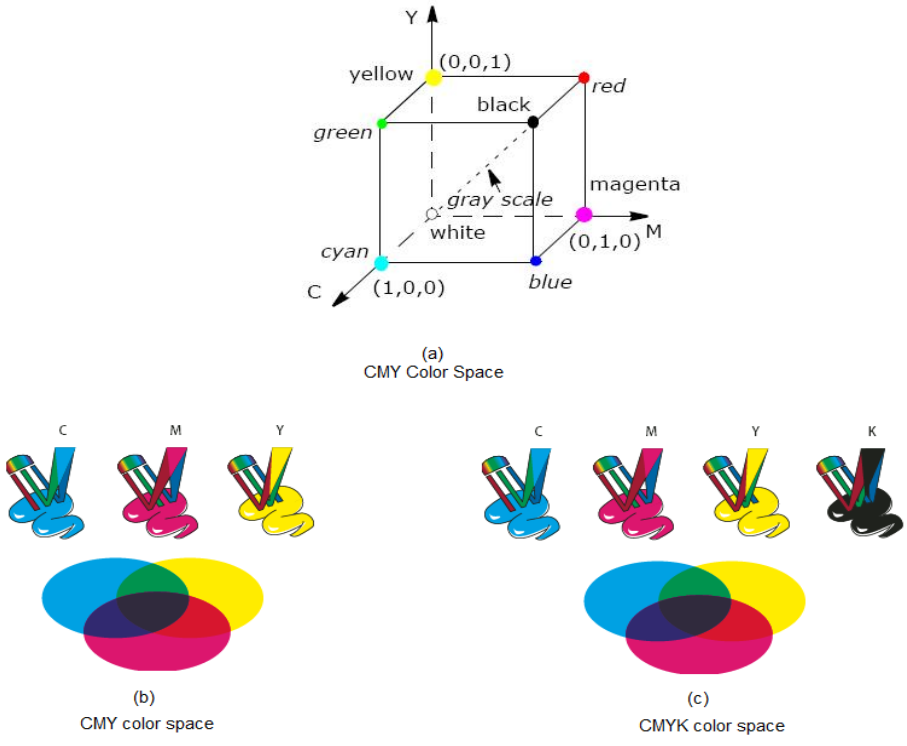


Figure 3.5: The CMY and CMYK color spaces; (a) CMY color space; (b) Generating colors in CMY color space (c) Generating colors in CMYK color space.

Most color printers and copiers that use pigments, require CMY data input. Since RGB model is the most commonly used model in computers, a conversion from RGB to CMY is needed. This conversion may be done by computers or internally by the printer or copier device. The conversion is performed using the following operation:

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{3.2}$$

However, actual printing with CMY inks generally cannot produce a very good black, but instead gives a muddy grayish brown. This reduces printing quality. Furthermore, mixing of the three colors to produce black would consume a large amount of colored ink. Sometimes, a combination of 100% cyan, magenta, and yellow inks soaks the paper with ink, making it slower to dry. The most common solution is to add a separate black ink to the printing process. Adding a fourth color, black (K), provides the CMYK color model as in Figure 3.5(c). This model reduces the need for large amounts of the colored ink, reduces the thickness of ink buildup on the page, makes the printing fast to dry, and reduces cost. From the standpoint of image quality, the most important factor is that it allows dark colors to be printed without appearing muddy (Gonzalez *et al.*, 2007).

3.6.3 YUV Model

Previous black-and-white TV systems used only (Y) information. The Y component, (luminance or luma) determines the brightness of a monochrome image that would be displayed by a black-and-white television receiver. Engineers needed a signal transmission method that was compatible with black-and-white (B&W) TV while being able to add color. Color information (U and V) was added separately so that a black-and-white receiver would still be able to receive and display a color picture transmission in the receiver's native black-and-white format. The YUV color model is used in the PAL and SECAM color TV broadcasting. Y ranges from 0 to 1 (or 0 to 255 in digital formats), while U and V range from -0.5 to 0.5, (or -128 to 127 in signed digital form, or 0 to 255 in unsigned form) as in Figure 3.6. Nowadays the term

YUV is commonly used in the computer industry to describe *file-formats* that are encoded using YCbCr.

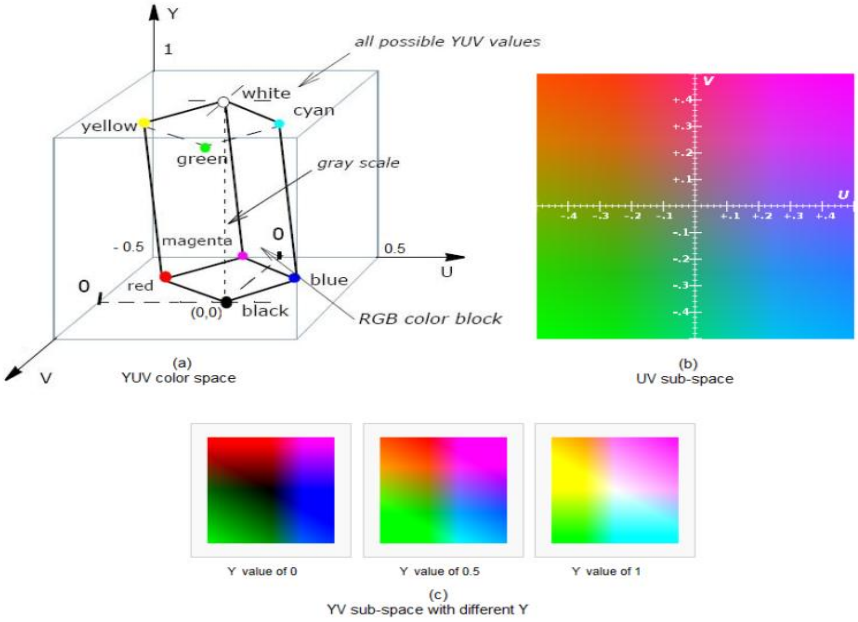


Figure 3.6: YUV color space; (a) YUV Color space; (b) UV sub-space; (c) UV sub-space with different Y, adopted from (Wikipedia).

For 8-bit (256 value) image, the conversion from RGB to YUV is (Russ, 2007):

$$Y = (0.299\ R) + (0.587\ G) + (0.114\ B);$$
$$U = 128 - (0.169\ R) - (0.331\ G) + (0.500\ B); \quad (3.3)$$
$$V = 128 + (0.500\ R) - (0.418\ G) - (0.081\ B)$$

This color space was used for skin color modelling and detection by (Li, Xue, & Fan, 2007) and (Vadakkepat *et al.*, 2008).

3.6.4 YIQ Model

YIQ was formerly used in the National Television Systems Committee (NTSC) television (North America, Japan, Thailand, and Korea). NTSC defines a color space known as YIQ which is similar to the YUV model. The YIQ and YUV stem from broadcast considerations (Russ, 2007). A main advantage of this format is that grayscale information is separated from color data, so the same signal can be used for both color and black and white sets. This system stores

a luminance value with two chrominance values, corresponding approximately to the amounts of blue and red in the color.

In the NTSC color space, image data consists of three components: luminance Y, which represents gray scale information, and IQ which make up chrominance (color information). The RGB to YIQ conversion is as follows (Russ, 2007):

$$\begin{aligned} Y &= 0.299\,R + 0.587\,G + 0.114\,B; \\ I &= 0.596\,R - 0.274\,G - 0.322\,B; \\ Q &= 0.211\,R - 0.523\,G + 0.312\,B \end{aligned} \tag{3.4}$$

The YIQ model was used for skin color modeling and detection by (Dai & Nakano, 1996), quoted by (Ruiz-del-Solar & Verschae, 2004).

3.6.5 HSV and HSI Models

The RGB, YUV and CMY color spaces are suitable for technical aspects. They do not correspond to the way that people recognize or describe colors. For example, one neither refers to the color of a car by giving the percentage of mixing the three primary colors, red, green, and blue, nor the percentage of mixing three pigments, cyan, magenta, and yellow. Usually, humans tend to describe the color of an object by its hue, saturation, and intensity such as blue, dark blue, etc.

HSV Model (Hue, Saturation, Value), also known as HSB (Hue, Saturation, Brightness) is more natural when thinking about a color and it is often used by people, image processing developers, and artists specifically, as it is closer to the way in which humans describe colors. HSV color space is very different from RGB or CMY. Figure 3.7 illustrates a representation of HSV. The cone shape has one axis running down its center, representing value V channel. If the cone is viewed from above, looking at its widest end, it becomes a circle with different colors around it.

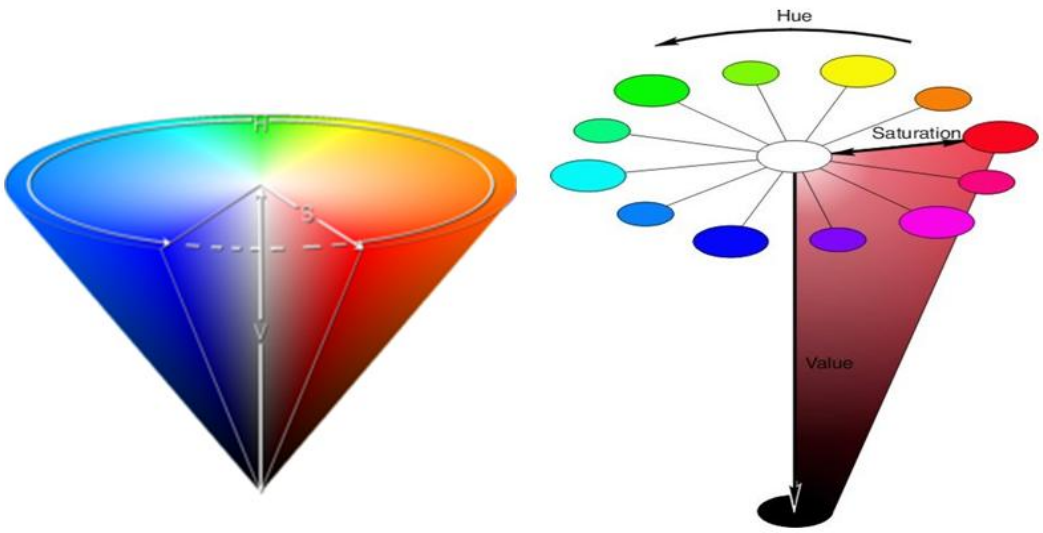


Figure 3.7: Perceptual representation of the HSV color space with the hue H (or θ) varies along the circumference; the saturation S varies along the radius, and the value V represents the brightness variations along the vertical axis.

The HSV color components are:

- **Hue** *H*, means the color itself (e.g. red, yellow, or violet). It is a measure of the spectral composition of a color. The graphical representation of hue is determined by an angular measurement analogous to a location around a color wheel (i.e. 0˚ to 360˚). A hue value of zero indicates the color red. The color green and blue correspond to 120° and 240° respectively, and then wrapping back to red at 360°, as shown in Fig 3.7.
- **Saturation** *S* refers to the purity of a color. On the outer edge of the hue wheel are the 'pure' hues (i.e. pure colors). As moving into the center of the wheel, the hue used to describe the color dominates less and less. At the center of the wheel, no hue dominates (i.e. colorless). In terms of a spectral definition of color, saturation is the ratio of the dominant wavelength to other wavelengths in the color. White light is white because it contains an even balance of all wavelengths. The value of saturation ranges from 0 to 1. A saturation of 1 (or 100%) means full pure color (i.e. colorfulness).
- **Value** *V* refers to how light or dark a color is (also referred to as lightness *L*, brightness *B*). V ranges also from 0 to 1. In terms of a spectral definition of color, V describes the overall intensity or strength of the light. While hue is a dimension going around a wheel, value *V* is a linear axis running through the middle of the wheel. The central

vertical axis comprises the gray colors, ranging from black at value = 0, the bottom, to white at value= 1, the top.

HSV color space which is closely related to the artist's concept of color, shade, and tone has many advantages for image processing and for understanding color. For instance, if the algorithms such as spatial smoothing or median filtering are used to reduce noise in an image, applying them to the RGB signals separately will cause color shifts in the result, but applying them to the HSV components will not (Russ, 2007).

The RGB to HSV conversion is defined by the following equations where *MAX* and *MIN* represent the maximum and minimum values of each R'G'B' triplet, where *R'=R/255; G'=G/255; B'=B/255* (MATLAB, 2010); (H. E. Burdick, 1997).

$$H = \begin{cases} \left(\frac{G' - B'}{MAX\text{-}MIN}\right)/6, & if \quad R' = MAX \\ \left(2 + \frac{B' - R'}{MAX\text{-}MIN}\right)/6, & if \quad G' = MAX \\ \left(4 + \frac{R' - G'}{MAX\text{-}MIN}\right)/6, & if \quad B' = MAX; \end{cases} \tag{3.5}$$

$$S = \frac{MAX\text{-}MIN}{MAX};$$

$$V = MAX$$

The HSI color space (hue, saturation, intensity) also known HSL (hue, saturation, lightness/luminance) is quite similar to HSV, with intensity I replacing the Value V component. The difference is that the brightness of a pure color is equal to the brightness of white, while the lightness of a pure color is equal to the lightness of a medium gray.

These color models were used for skin color modeling and detection by (Adipranata, Ballangan, & Ongkodjojo, 2008; Baskan, Bulut, & Atalay, 2002; Garcia & Tziritas, 1999; Juang & Shiu, 2008; McKenna, Gong, & Raja, 1998; Moallem et al., 2011; Sandeep & Rajagopalan, 2002; Tomaz et al., 2004; Zainuddin & Naji, 2010)

3.6.6 CIE Model

CIE model stands for Commission Internationale de L'Eclairage (the International Commission on Illumination). The commission was founded in 1913 as an autonomous international board to provide a forum for the exchange of ideas and information and to set standards for all things related to lighting. Later this model was developed in 1931 to be completely independent of any device. The CEI XYZ chromaticity diagram is a two-dimensional plot defining colors (Russ, 2007) as in Figure 3.8 which shows that the colors are fully saturated along the edge. The two axes, X and Y, are always positive (unlike the U and V values). Numbers give the wavelength of light in nanometers. The inscribed triangle shows the colors that typical color CRTs can produce by mixing red, green, and blue light from phosphors. The third (perpendicular) axis Z is the luminance, which corresponds to the brightness which, like the Y value in YUV, produces a monochrome (gray scale) image. From this model, other models were derived in response to various concerns such as CIE LAB 1942, CIE LUV 1960, and CIE L*a*b* 1976. The CIE diagram provides a tool for color definition, but corresponds neither to the operation of hardware nor directly to human vision (Russ, 2007).

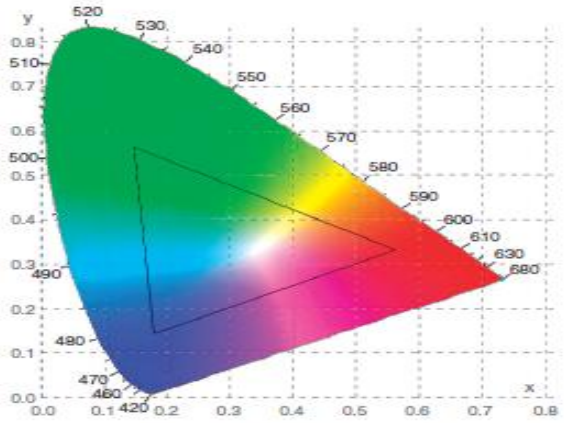


Figure 3.8: The CIE chromaticity diagram, adopted from (Russ, 2007).

The transformation from RGB to CIE XYZ model is performed as follows (Chaves-González *et al.*, 2010):

$$\begin{aligned} X &= 0.490186\,R + 0.309879\,G + 0.199934\,B; \\ Y &= 0.177015\,R + 0.812324\,G + 0.010660\,B; \\ Z &= 0.010077\,G + 0.989922\,B \end{aligned} \tag{3.6}$$

where all the components (R, G, B, X, Y and Z) are in the range from 0 to 1.

These color models were used for skin color modeling and skin detection by (M. H. Yang & Ahuja, 1998) and (Q. Chen, Wu, & Yachida, 1995).

3.6.7 YCbCr Color Space

YCbCr color space is widely used for digital video. In this format, luminance information is stored as a single component (Y), and chrominance information is stored as two color-difference components Cb and Cr (very similar to YUV but not identical). The Cb represents the difference between the blue component and a reference value as in equation (3.7). The Cr represents the difference between the red component and a reference value. YCbCr data can be double precision, but the color space is particularly well suited to unsigned integers 8-bits data (i.e. uint8). For Maltab images, the data range for Y is [16, 235], and the range for Cb and Cr is [16, 240] (MATLAB, 2010). YCbCr leaves room at the top and bottom of the full uint8 range so that additional (nonimage) information can be included in a video stream.

The conversion from RGB to YCbCr is simply (Vezhnevets *et al.*, 2003):

$$\begin{aligned} Y &= 0.299\,R + 0.587\,G + 0.114\,B; \\ Cr &= R - Y; \\ Cb &= B - Y \end{aligned} \tag{3.7}$$

Due to the simplicity of this transformation and explicit separation of luminance and chrominance components, many researchers used this color space for skin color modeling and image segmentation (Chai & Ngan, 1999; Chai, Phung, & Bouzerdoum, 2003; Frisch *et al.*, 2007; Garcia & Tziritas, 1999; Ghazali *et al.*, 2012; Habili *et al.*, 2004; Hsu *et al.*, 2002; Kumar

& Bindu, 2006; Menser & Wien, 2000; Phung, Bouzerdoum, & Chai, 2003; Shih *et al.*, 2008; Zaidan, Ahmed, Karim, Alam, & Zaidan, 2010).

3.6.8 Comparison of Color Spaces for Skin Detection

Many comparative studies on the skin color modeling using different color spaces are reported in the literature. Zarit, Super, and Quek (1999) performed a comparative evaluation of pixel-based skin detection performance in five color spaces, these are: CIE Lab, HSV, Normalized RGB, YCbCr, and Fleck HS space. They used two methods: a lookup-table and a Bayesian decision theory. The methods were tested with different images downloaded from a variety of sources to include a wide range of skin tones, environments, cameras, and lighting conditions. They report that lookup-tables for HSV color space show the best performance. Considering Bayesian method, the choice of the color space had no significant difference in the results.

Terrillon *et al.* (2000) did a similar study to compare the efficiency of nine different color spaces for skin detection against complex backgrounds, these are: Normalized RGB, CIE-xyz, HSV, YIQ, TSL, CIE-DSH, YES, CIE Luv, and CIE Lab. They modeled the skin distribution as a single Gaussian model based on the Mahalanobis metric and a Gaussian mixture density model, respectively. To the best of our knowledge, The TSL color space is not a standard color space and it was devised and used by the authors. The authors reported that their normalized TSL space (i.e. TS) yielded the best segmentation results.

Albiol, Torres, and Delp (2000) argued that there is an optimum skin model for every color space, i.e., the classification accuracy is independent on the color space. They demonstrated this theoretically for three color spaces: RGB, HSV, and YCbCr. No quantitative results were given concerning the dataset used in this work.

Shin, Chang, and Tsap (2002) have evaluated eight color spaces to find which color space setting is most suitable for skin detection. These are: normalized RGB, CIE XYZ, CIE lab,

HSI, SCT, YCbCr, YIQ, and YUV. They examined if the color space transformation does improve the detection rate by measuring four separability measurements on a large dataset of 805 images with different skin tones and illumination. The authors argued that the color space transformations did not improve the performance in the task of skin detection. They found better discrimination of skin and non-skin pixels in RGB color space.

Vezhnevets *et al.* (Vezhnevets *et al.*, 2003) conducted a survey on pixel-based skin color detection techniques. They determined that ignoring the darkness component (i.e. color luminance) does not improve the discrimination of skin and non-skin pixels, but it helps to generalize sparse training data.

Schmugge *et al.* (2007) compared the performance of nine color spaces with the presence or the absence of the luminance component using two color modeling approaches for different settings (indoor or outdoor) and modeling parameters. The performance is measured by using a receiver operating characteristic (ROC) curve on a large dataset of 845 images (consisting more than 18.6 million pixels) with manual ground truth. The authors concluded that 1) the color space transformation does improve the performance, but not consistently, 2) ignoring the luminance component decreases performance, 3) the best performance was obtained using HSI or SCT color spaces, keeping the luminance component, and modeling the color with the histogram approach.

Chaves-González *et al.* (2010) performed a recent study to determine which color model is the best option to build an efficient skin detector. They studied 10 of the most common used color spaces, and doing different comparisons. According to their results, the most appropriate color space for skin color detection is the HSV model. This study agrees with the HSV properties that are in most of the text books and literatures in the field of computer vision and digital image processing such as (H. E. Burdick, 1997; Efford, 2000; Gonzalez & Woods, 2002).

As discussed in the previous sections, each model is oriented toward some hardware or application needs. According to our knowledge, there is no single color model that can surpass others for segmenting all kinds of color images.

3.7 Skin Color Modeling – Literature Investigation and Discussion

"The term ***model*** reflects the fact that any natural phenomenon can be described to a certain degree of accuracy and correctness. Therefore, it is important to seek through all natural sciences for the simplest and most general description that still describes the observations with minimum deviations" (Jahne, 2005). It is the power and attractiveness of the basic laws of mathematics, statistics, logic, physics, and human experiments that a phenomenon can be described on the basis of a few simple and general principles. It is the correct approach for a certain task such as human skin segmentation to look for the simplest skin color clustering model that describes this task in the most accurate way. Actually, an analysis of statistical measurements and error rate relies completely on the skin color model assumptions and it is only valid as long as the model agrees to the experimental setup. Even if the results seem to be good with the skin color model assumptions, there is no guarantee that the model's assumptions are correct or it is the most excellent model. This is because different models may produce similar results. For the image segmentation problem, skin color modeling method will help to discriminate the human skin and non-skin pixels based on color information. This means that skin regions are detected by looking for pixels that have color that correspond to (or matched with) skin model (Phung *et al.*, 2003).

In this section, we review some existing methods for modeling skin color used for classification and segmentation. These methods are classified into three categories as in Table 3.1 (Vezhnevets *et al.*, 2003).

Table 3.1. Summary of main categories used for skin color modeling methods

Explicit defined skin-color thresholding	This category uses some explicit rules to define the skin color clustering models based on threshold principles. Therefore, single or multiple levels of thresholds are set for each color component.
Non-parametric	The key idea of the non-parametric skin modeling methods is to estimate skin color distribution from the training data without deriving an explicit model of the skin color.
Parametric	This category uses parametric models for skin color distributions. This model usually consists of a Gaussian or a mixture of Gaussian models to describe the skin color distribution in different color spaces.

3.7.1 Explicit Defined Skin Color Thresholding

Thresholding is the simplest method of image segmentation that enjoys a significant degree of popularity. It can be regarded as the fastest method for separating objects from their surroundings (Shapiro & Stockman, 2001). During the thresholding process, individual pixels in an image are marked as "object" pixels if their values are in the range of threshold values and as "background" pixels otherwise. Typically, an object pixel is given a value of “1” while a background pixel is given a value of “0” (creating binary images).

The thresholding process depends mainly on selecting the correct threshold value (or values). For selecting threshold values many methods are proposed. Users can manually choose a threshold value and interactively see the results. Trial and error comes into play and the result is as good as you want it to be. Other methods use a thresholding algorithm to compute the value automatically, which is known as automatic thresholding. For example, computing the mean and variance is a simple method to estimate thresholding values. Another widely used approach is to create a histogram of the image and select the valley points as the thresholds.

Color images are segmented by designating a separate threshold value(s) for each color component (Russ, 2007). In other words, there are multiple thresholds. The skin regions are detected as follows: Pixels with color features falling within the ranges of these threshold levels would be classified as skin pixels or otherwise belong to the background.

For example, by using three R, G, B histograms in RGB space, one can choose three threshold levels to select pixels that lie within a portion of the color space that is a small sub-cube, as shown in Figure 3.9, adopted from (Russ, 2007). The figure shows the combination of separate thresholds on each individual color component. The three threshold levels are combined with a logical Boolean AND operator to generate one conditional rule. In this figure, the shaded area is the Boolean AND of the three threshold settings. The limitation of this method is clear - the only shape that can be formed in 3D space is a rectangular prism.

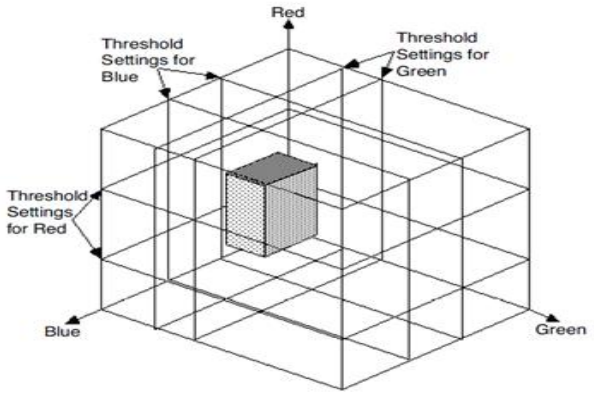


Figure 3.9: Explicit defined threshold values on individual color channels. The shaded area is the Boolean AND of the three threshold settings for RGB, adopted from (Russ 2007).

This method can be extended by imposing additional explicit defined skin region thresholds. A good example of explicit defined skin region is implemented by (Peer & Solina, 1999; Solina, Peer, Batagelj, & Juvan, 2002) as shown in equation (3.8). An image pixel is classified as skin when the following conditions are hold:

$$\begin{aligned} &R > 95 \text{ and } G > 40 \text{ and } B > 20 \text{ and;} \\ &Max(R, G, B) - min(R, G, B) > 15 \text{ and;} \\ &|R - G| > 15 \text{ and } R > G \text{ and } R > B \end{aligned} \tag{3.8}$$

Chen and Wang (2007) used a set of threshold rules that was empirically constructed in the RGB color space.:

$$\begin{aligned} &R > G \text{ and } G > B \text{ and;} \\ &R > 95 \text{ and } G > 40 \text{ and } B > 20 \text{ and;} \\ &30 < (R - G) < 80 \text{ and } (R - B) < 120 \text{ and;} \\ &10 < (G - B) < 80 \text{ and } (G + B - R) > 10 \end{aligned} \tag{3.9}$$

In the research by Baskan *et al.* (2002) two skin color filters in HSV color space are used based on HS only (i.e. the illumination component V was ignored) as follows:

- Skin Filter-1 is designed to extract the skin colored regions from the image using the following thresholds:

$$0.23 \leq S \leq 0.69, \quad \text{and} \quad 0° \leq H \leq 40° \tag{3.10}$$

  where S indicates the saturation component and H the hue component of Hue-saturation-intensity representation of color.

- The second filter, Skin Filter-2 is designed with the following thresholds:

$$0.23 \leq S \leq 0.69, \text{ and } 0° \leq H \leq 40° \text{ , and } S' \geq 0.25 \tag{3.11}$$

where S' corresponds to the saturation value of the pixel of the negative image.

For a source image, both filters are applied and the one, which gives the better shape of the face, is selected. They assume that:

- The background is not complex.
- There is only a single face in an input image.
- The image quality and resolution is sufficient enough.
- The illumination is uniform and the input images are color images. However, no restrictions on clothes, glasses, makeup, hairstyle, beard, etc. are imposed.
- They use ellipse fitting technique that is the oval shape of the segmentation output.

These precondition and assumptions limit the usage of this approach and also, the performance of such a system is not stable since usually images come from different sources and types.

Sobottka & Pitas (1998) also considered that hue and saturation HS are sufficient to discriminate color information for segmentation of skin regions (i.e. without taking the intensity value V into account). Based on extensive experiments, the thresholds rules that are used for skin detection are:

$$0.23 \leq S \leq 0.68 \text{ and } 0° \leq H \leq 50 \tag{3.12}$$

These values have been determined using training pixels collected from the M2VTS database, containing images of yellow and white skinned people. The graphical representation of these rules would be equivalent to a sector at the HSV color space as shown in Figure 3.10. The shaded region defines the skin color cluster.

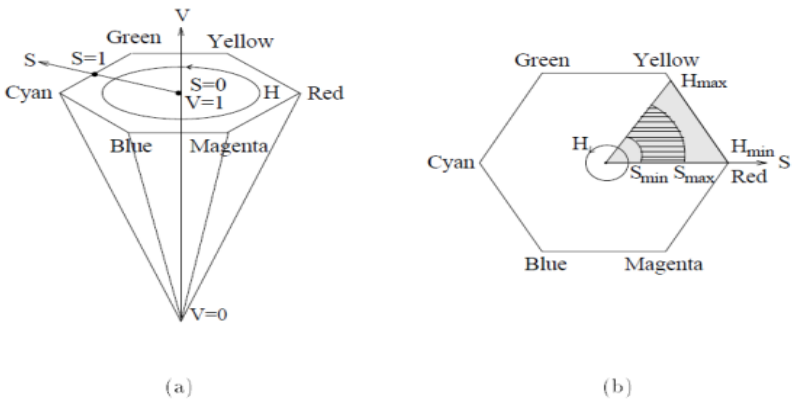


Figure 3.10: The graphical representation of classification rules used by Sobottka (1998).

Do, You, & Chien (2007) used HSV color space with the following decision boundaries for skin color detection:

$$0° < H < 50° \text{ and } 0.20 < S < 0.68 \text{ and } 0.35 < V < 1.0 \tag{3.13}$$

Garcia and Tziritas (1999) also used this method and reported the equations for defining six bounding planes that have been found by successive adjustments according to segmentation results in the HSV color space:

$$
\begin{aligned}
&S \geq 10;\ V \geq 40;\ \ S \leq -H - 0.1\,V + 110\ ;\\
&H \leq -0.4\,V + 75 \text{ and;}\\
&\text{If } H \geq 0; \qquad\qquad (3.14)\\
&\qquad S \leq 0.08(100-V)\,H + 0.5\,V;\\
&\text{else}\\
&\qquad S \leq 0.5\,H + 35
\end{aligned}
$$

The intersections of the adjusted bounding planes with the HS plane for V=70 are drawn as shown in Figure 3.11. However, they noticed that the bounding planes are more easily adjusted using the HSV than YCbCr model, because of a direct access to H (Hue) which mainly encodes skin colors.

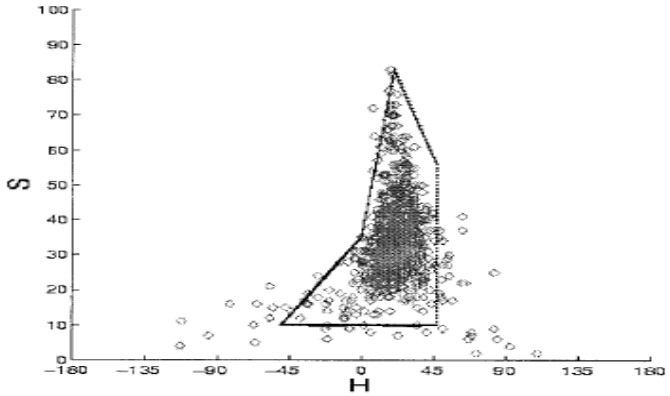


Figure 3.11: The bounding planes with the HS plane for v=70 used by Garcia (1999).

Garcia *et al.* (1999) also used YCbCr color space to define skin color space. They used these rules:

$$
\begin{aligned}
&\text{If } (Y > 128)\\
&\qquad \theta_1 = -2 + (256-Y)/16;\\
&\qquad \theta_2 = 20-(256-Y)/16;\\
&\qquad \theta_3 = 6;\\
&\qquad \theta_4 = -8;\\
&\text{Else}\\
&\qquad \theta_2 = 12;\\
&\qquad \theta_3 = 2+Y/32; \qquad\qquad (3.15)\\
&\qquad \theta_4 = -16+Y/16;\\
&\text{end;}\\
&Cr >= -2*(Cb+24);\ Cr >= -(Cb+17);\ Cr >= -4*(Cb+32);\ Cr >= 2.5*(Cb+\theta 1);\\
&Cr >= \theta 3;\ \ Cr >= 2.5*(\theta 4-Cb)\ ;\ \ Cr <= (220-Cb)/6\ \ ;\ \ Cr <= 4/3*(\theta 2-Cb)
\end{aligned}
$$

Chai & Ngan (1999) found that a skin-color region could be detected by the presence of a certain set of chrominance (i.e., Cb and Cr) values narrowly and consistently distributed in the YCbCr color space. The pixel values in the range Cb = [77, 127], and Cr =[133, 173] are defined as skin pixels based on data samples from the ECU face and skin database. The researcher ignores the illumination component Y.

Mixed color spaces of the normalized RGB model and the HSV model were used by Wang & Yuan (2001). They chose the two-model parameters as follows:

$$\begin{aligned} &0.36 < R < 0.465 \;\; and \;\; 0.28 < G < 0.363; \\ &0° < H < 50° \;\; and \;\; 0.20 < S < 0.68 \;\; and \;\; 0.35 < V < 1.0 \end{aligned} \tag{3.16}$$

Vadakkepat *et al.* (2008) used the following rules using two color spaces YCbCr and YUV but again the intensity component Y was ignored:

$$\begin{aligned} &138 < Cr < 178; \\ &200 < Cb + 0.6\,Cr < 215; \\ &\\ &-30 < U < 5; \\ &-4.2 < V < 28.8; \\ &V < 30; \\ &U > 0.45\,V - 37.65\,; \\ &U > -2.37\,V - 17.65; \\ &V < 30; \\ &U < 0.206\,V + 2.94; \\ &U > 0.08\,V^2 - 2.4\,V - 17.2 \end{aligned} \tag{3.17}$$

.

Adaptive thresholding was proposed by Cho, Jang, and Hong (2001). They used this method to detect skin color regions in a color image by adaptively adjusting the threshold values. The initial upper and lower threshold values for each color component are H=[0.4,0.7], S=[0.15,0.75], V=[0.35,0.95]. Then, the threshold values for S and V components are updated iteratively based on a color histogram built in SV space. However, the proposed method implies many assumuptions and preconditions (e.g. single face and dominant color) that makes it applicable for limited applications.

The adaptive thresholding technique was also used by (Soriano, Martinkauppi, Huovinen, & Laaksonen, 2003) using normilzed RG space. The proposed approach updates a dynamic skin color model under changing illumination conditions.

3.7.2 Nonparametric skin distribution modeling

The idea of the non-parametric skin modeling methods is to build a color-based classifier using training data without building a specific model. The sections below cover the methods of Distance-based segmentation, Lookup-tables, Bayes classifier, Fuzzy Logic, Neural Networks, and Support Vector Machines.

3.7.2.1 Distance Based Segmentation

One of the most intuitive measures of similarity of colors in the color space is the *Euclidean distance* (Gonzalez *et al.*, 2007). The idea of pattern classification by distance is based on a simple heuristic: similar colors appear closer to each other in the color space. By collecting a set of sample skin color pixels as representative of the color of interest (i.e. human skin patches), we compute the "average" or "mean" color *M* of all points that we wish to detect or segment. If we consider M as a center point of a sphere S with radius T, then we can use T as a threshold to measure the similarity of color, as in Figure 3.12(a), adopted from (Gonzalez *et al.*, 2007). The goal is to classify an unknown pixel P as having a color like skin color M or otherwise. We say that P is similar to M if the *Euclidean distance* $D(P,M)$ between them is less than or equal to T. Points lying within the sphere S would be classified as skin pixels; points outside the sphere would be classified as non-skin pixels.

For example in the RGB model *Euclidean* distance between P and M is as follows:

$$\begin{aligned} D(P,M) &= \|P-M\| \\ &= \left[(P-M)^T(P-M)\right]^{1/2} \\ &= \left[(P_R-M_R)^2+(P_G-M_G)^2+(P_B-M_B)^2\right]^{1/2} \end{aligned} \tag{3.18}$$

where ||.|| is the norm of the arguments and subscripts R,G, and B denote the color components of vectors P and M.

The main drawback of this technique: It classifies the pixels of an image based on the distance (i.e. *Euclidean* distance) between their feature vectors without considering the global distribution of a feature. As a result, artifacts are likely to occur in the segmentation (Aghbari & Al-Haj, 2006). A good solution is to use *Mahalanobis* distance that considers the direction of data spread. Hence, the classification of points is enclosed by an ellipsoid body instead of a sphere as in Figure 3.12(b). The *Mahalanobis distance* from the P color vector to mean vector M, given the covariance matrix C of the samples, can serve this purpose (Terrillon *et al.*, 2000):

$$D(P,M) = \left[(P-M)^T C^{-1}(P-M)\right]^{1/2} \tag{3.19}$$

Eq. 3.19 defines elliptical surface in chrominance space centered about M and whose principal axes are oriented in the direction of maximum data spread.

Distance based segmentation can also be used for multi-class classification problems. That is, a point P belongs to class ω_i on the basis that it is closer to the samples of this class, in terms of minimum-distance.

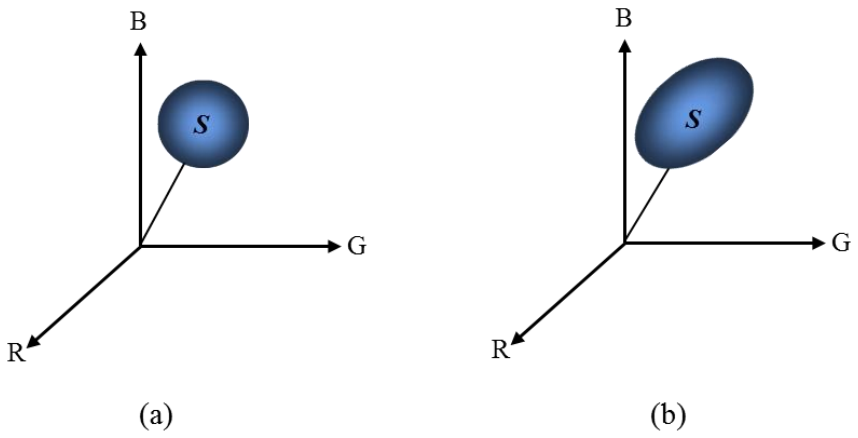


Figure 3.12: Two distance-based approaches for clustering skin color in RGB model for the purpose of skin segmentation; (a) *Euclidean* distance. (b) *Mahalanobis* distance.

3.7.2.2 Lookup-Tables (LUT)

An approach for skin color segmentation without arithmetic operations is the so-called Lookup tables which are constructed offline and indexed by a color information vector, and each cell contains information on whether the indexed color belongs to the skin or non-skin class. The

process should pass over all pixels in the source image. Lookup table based on histogram is a commonly-used approach. After collecting the training data (i.e. skin pixels) as representative of the colors of interest, each entry of the histogram counts how many times a particular color occurs in the training data. Then, the histogram is converted to discrete probability distribution to be normalized. The normalized lookup-table represents the likelihood values that a specific color will correspond to skin color. Wu , Chen, and Yachida (1999) used UCS color system (which is transformed from CIE XYZ color space) to construct the histogram.

Dynamic histograms proposed by Sigal, Sclaroff, and Athitsos (2000) and Soriano *et al.* (2000) are updated based on feedback from the current segmentation.

Zaqout *et al.* (2004) used three Lookup tables. The entries in the LUT represent the frequency of color pixels that fall in a particular range, that is, the occurrence proportion and certainty value. They start by creating three LUTs based on the relationship between each pair of the triple components, (namely, G:R, B:R, and B:G) from their histograms.

The relationship between lookup tables and choosing the right color space was investigated by a comparative study of Zarit, Super, and Quek (1999). The authors carry out this study across five color spaces.

3.7.2.3 Bayes Classifier - Statistical Approach

Gomez, Sanchez, and Enrique Sucar (2002) proposed a simplified version of probability theory using RGB color space. Initially, 3D histograms are constructed. Then, the conditional probability of a pixel with RGB value to be skin or non-skin is:

$$P(\text{rgb}|\text{skin}) = \frac{\text{Hist}_{\text{skin}}[\text{r,g,b}]}{\text{Total}_{\text{skin}}}, \quad P(\text{rgb}|{\sim}\text{skin}) = \frac{\text{Hist}_{\text{non-skin}}[\text{r,g,b}]}{\text{Total}_{\text{non-skin}}} \tag{3.20}$$

A new pixel can be labeled as skin if it satisfies a given threshold θ:

$$\frac{P(\text{rgb}|\text{skin})}{P(\text{rgb}|{\sim}\text{skin})} \geq \theta \tag{3.21}$$

where θ is obtained empirically. It is clear that there are a number of issues such as generalization of data, size of data, noise, and prior probabilities that affect the overall detector behavior.

The Bayes classifier with histogram technique has been used for skin detection by Jones and Rehg (2002); Chai *et al*. (2003); Phung, Bouzerdoum, and Chai (2005); Ma and Leijon (2010).

3.7.2.4 Fuzzy Logic

Kim et al. (2005) used fuzzy skin color model for the detection of skin regions from images based on fuzzy inference rule-based system. The three color components of HSI color space is used in these rules. The clustering method is based on the membership functions in each rule which are treated as fuzzy cluster function. This clustering method corresponds to the probability value of cluster as output of firing strength instead of simple fuzzy set. The authors did not provide any details about the experimental results and the detection rate of the system.

Moallem et al. (2011) proposed Fuzzy Inference Systems (FIS) for skin segmentation based on Euclidean distance, Fuzzy rules, and genetic algorithms (GA). They used more than one million pixels gathered from skin samples of different face databases. First, by using HSI color space, in which the average of the chosen color space is computed as the skin vector mean. After transforming the input image into the chosen color space, the fuzzy system is used with 1-input, 1-output. The system then applied the normalized Euclidean distance between the color of each new pixel and the skin vector mean as an input, and the likelihood of being skin pixel as an output. Subtractive clustering was applied on input space (containing 132,000 skin and non-skin pixels) to decide on the number of membership functions (MF's) and rules. Utilizing the four clusters information and experimental knowledge, input and output MF's were designed. A semantic meaning for each cluster was used for better understanding (i.e. Skin, Rather Skin, Low Probability Skin, Non-Skin). The achieved rule in skin-color segmentation FIS is:

IF input is Z, THEN output is Z

where Z ∈ [Skin, Rather Skin, Low Probability Skin, Non-Skin].

The result of applying such a system is the skin-likelihood image, that is, the gray scale image whose gray values represent the likelihood of the pixel belonging to the skin. To make a binary image, an appropriate threshold should be selected, which is optimized by GA. The threshold is the chromosome of the GA, whose fitness function compared the whole detected skin pixels in the sample images with the actual number of these pixels, and attempted to minimize the difference. However, no quantitative results on skin detection were presented.

3.7.2.5 Neural Networks for skin segmentation

In multilayer perceptron-based skin classification, a neural network (NN) is trained to learn the complex class conditional distributions of the skin and non-skin pixels.

Phung, Chai, and Bouzerdoum (2001) presented NN-based skin color detector using YCbCr color space but ignored the Y component. The steps in their NN detector include presenting a neural network with a number of training pairs, each of which consists of a feature vector $[Cb\ Cr]^T$ and a corresponding class indicator. Then the network's parameters were adjusted through supervised training to produce the expected class indicators for the given feature vectors. The best result was 91.6% correct classification, achieved with a neural net of size 2-25-1 (one hidden layer of 25 neurons with activation function logsig) and with output threshold $\theta = 0.3$.

Brown, Craw, and Lewthwaite (2001) described a method of skin detection using a Self-Organising Map (SOM) which achieved consistent accuracy of over 94%.

Seow, Valaparla, and Asari (2003) presented a NN-based skin color detector. The primary color components of each plane of the RGB color cube are fed to a three-layered network. The NN-based classifier was trained using the backpropagation algorithm with the training samples, to

extract the skin regions from the planes and interpolate them to provide an optimum decision boundary and hence the positive skin samples for the skin detector.

Sebe *et al.* (2004) proposed a Bayesian network classifier for skin detection using RG chromaticity space. They tested the performance of three classifiers: Naive Bayes (NB), Tree-Augmented Naive Bayes (TAN), and driven Stochastic Structure Search (SSS). They reported that the classifier learned with the SSS algorithm shows the best results and outperforms both TAN and NB classifiers. The average detection rate with the false alarm of 1%, 5%, and 10% are 87.66, 95.82, and 98.32 respectively.

Taqa and Jalab (2010) proposed a skin detection method based on neural network classifier that combines both color and texture features. The classifier was trained using the color of a pixel and the neighbourhood pixel information (i.e. texture) which was calculated for this purpose. The texture includes range, standard deviation, and entropy. The system used RGB color space.

Zaidan *et al.* (2010) also proposed a hybrid module for skin segmentation using neural network as well as heuristic rules. The system used YCbCr color space and achieved detection rates of 88.5%.

3.7.2.6 SVM for skin segmentation

Han, Awad, and Sutherland (2009) proposed a skin detection approach in application to sign language recognition system consisting of two stages. First, a binary classifier based on SVM was trained using the sample of skin pixels. Then, active learning was employed to select the most informative training subset for SVM, which leads to fast convergence and better performance. Moreover, to reduce the noise and illumination variations the system used region-based information. Three metrics were employed to measure the performance, correct detection rate (CDR), false detection rate (FDR) and overall classification rate (CR) and these are 86.34, 0.96, 76.77 respectively.

3.7.3 Parametric Skin Distribution Modeling

3.7.3.1 Gaussian Distribution

The idea of pattern classification by Gaussian distribution is based on a simple heuristic that skin color distribution can be modeled based on Gaussian distribution in the color space. All Gaussian (Normal) distributions look like a symmetric, bell-shaped curve. The graph of the normal distribution depends on two factors: the mean μ and the standard deviation σ. The mean of the distribution determines the location of the center of the graph, and the standard deviation determines the height and width of the graph. The Probability Density Function (PDF) of a random variable x denoted by $P(x)$ is calculated as follows (Duda, Hart, & Stork, 2001):

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-1}{2} \left(\frac{(x-\mu)}{\sigma} \right)^2 \right) \tag{3.22}$$

where x is a normal random variable, μ is the mean, σ is the standard deviation, π is approximately 3.14159. This is called the standard normal distribution (Jain, 1989). The multivariate Gaussian distribution is a generalization of the one-dimensional standard normal distribution to higher dimensions. The general multivariate normal density in *d*-dimensions is given by (Duda *et al.*, 2001; Gonzalez *et al.*, 2007):

$$P(x \mid \omega_j) = \frac{1}{(2\pi)^{d/2} \mid C_j \mid^{1/2}} \exp\left(\frac{-1}{2} (x-\mu_j)^T C_j^{-1} (x-\mu_j) \right) \tag{3.23}$$

where x is a *d*-component column vector, C_j and μ_j are the *covariance matrix* and *mean vector* of the pattern population of class ω_j, $\mid C_j \mid$ is the determinant of C_j, and $(x-\mu)^T$ is the transpose of $(x-\mu)$.

Many of the representative works on skin-color distribution modelling have used Gaussian density functions and Gaussian mixtures (Shih *et al.*, 2008) (Caetano, Olabarriaga, & Barone, 2002) (Hsu *et al.*, 2002) (Z. Liu *et al.*, 2005). The iterative expectation-maximization (EM)

algorithm is widely used in many previous works for parameter estimation. A good description of the EM algorithm for parameter estimation and testing the goodness-of-fit of Gaussian mixture can be found in (M. H. Yang, 2000). The advantage of these parametric models is that they can generalize well with less training data and have much less storage requirements.

Yang and Ahuja (1998) used CIE LUV color space and discarded the luminance value L. The distribution of skin color is expressed by $x=(U,V)^T$, and modelled by a Gaussian distribution. Therefore, they hypothesized the distribution of skin color as a bivariate Gaussian distribution $N(\mu,\sum)$ where $\mu = (\mu_u, \mu_v)$ and

$$\sum = \begin{vmatrix} \sigma_{uu}^2 & \sigma_{uv}^2 \\ \sigma_{vu}^2 & \sigma_{vv}^2 \end{vmatrix} \tag{3.24}$$

A pixel is classified as skin-like color if its corresponding probability is greater than a threshold T where T=0.5 and a region is identified as a human skin color if most (above 70%) of its pixels have skin color. Extensive experiments reveal that a mixture model (GMM) gives better results than a unimodal Gaussian (or Single Gaussian Model SGM). The method is tested on a large dataset. However, no quantitative results on skin detection were presented.

Shih *et al.* (2008) randomly selected 80 color face images from the Internet and extracted the face skin patch of size 20×20 each to establish the 2-D Gaussian skin-color model in Y*CbCr* color space (ignoring the Luminance Y component). The parameters are calculated using the maximum likelihood method as follows:

$$\mu = \begin{bmatrix} Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 116.88 \\ 158.71 \end{bmatrix}, \quad \sum = \begin{bmatrix} 74.19 & -43.73 \\ -43.73 & 82.76 \end{bmatrix}, \quad \rho = -0.5581 \tag{3.25}$$

where $\boldsymbol{\mu}$ is mean vector, $\boldsymbol{\Sigma}$ is covariance matrix, and ρ is the correlation coefficient.

A probability value is calculated for each input pixel in the source image to indicate its likelihood to be skin pixel. The skin-likelihood probabilities for the whole image are normalized in the range of [1, 100]. Then, a threshold-based approach is used to segment the likelihood image into skin and non-skin regions.

Liu *et al.* (2005) used a 2D Gaussian Probability Density Function (PDF) to model the skin color distribution in YUV color space, where the chrominance vector is $x=[U\ V]^T$ and the mean vector μ_s and the covariance matrix $\sum_s$ of the skin class are estimated from a training set. Then the watershed segmentation algorithm is used to partition the source image into a set of regions. The Boolean region adjacency graph RAG is then generated to reflect the adjacent relationship of different segmented regions to produce a labeled image. Skin-like regions can be obtained using the result of pixel classification and the label image. The authors collected more than 2,500,000 skin pixels from people of several ethnic groups, but they ignored the first component Y, the brightness of the color.

Although skin colors of different races fall into a small cluster in normalized RGB or HSV color space, Yang (2000) and Greenspan, Goldberger, and Eshet (2001) found that a single Gaussian distribution is neither sufficient to model human skin color nor effective in general applications. Furthermore, previous approaches used small collections of images to estimate the density function but did not validate the models by verifying the statistical fit of the chosen model to the data. Greenspan *et al.* (2001) provided a statistical test to show that a Gaussian mixture model provides a more robust representation that can capture multiple variation of skin color and variations in lighting conditions. They used normalized *R-G* color space and a large set of training samples from ARH and ARL database.

Caetano *et al.* (2002) showed that mixture models can improve skin detection, but not always. The authors used a single Gaussian model which is estimated analytically via the maximum-likelihood (ML) criterion and seven mixture models (from 2 to 8 Gaussians) which are estimated via the expectation-maximization EM algorithm. They used *RG* color space on images containing both black and white people. The performance evaluation was applied on a test set containing 10,608,076 skin pixels and 440,451,063 non-skin pixels. The conclusions driven by the authors can be summarised as follows: First, the single Gaussian model shows poor performance. Second, all the Gaussian Mixtures show similar performance over the whole

range of possible operating points. Consequently, mixture models are not necessarily the best option for skin color modelling in RG space, but just under the special condition of high TPRs.

Lee and Yoo (2002) discussed the limitations of the Gaussian models and suggested a new statistical color model for skin detection, called an elliptical boundary model. They compared their method to those of the single and mixture of Gaussian model. Each model was trained and tested using images from Compaq database. The authors argued that the elliptical boundary model can be easily constructed from training data in a fast speed and its performance is better than both the single and the mixture of Gaussian model.

Jones and Rehg (2002) compared the performance of histogram with Gaussian mixture models for skin detection. They collect about 1 billion skin pixels and conducted extensive experiments using RGB color space. They found the histogram models to be superior in accuracy and computational cost.

Yuetao and Nana (2011) used a mixed skin color model that combined YCbCr and normalized RGB models. According to the two dimensional skin Gaussian model formula, the system transforms the source color image into likelihood image based on pixel color properties x = [Cr,Cb]. The gray value corresponds to the possibility of that point belonging to the skin region (the lighter the gray, the closer to the skin color). Then, thresholding technique is used with RG space such that a pixels color *R* in range [0.36, 0.51] and *G* in range [0.28, 0.35], and *R*>*G*, is classified as a skin pixel, otherwise it is not. Then, the system combines the binary images of the two generated images with logical *AND* to obtain the final skin segmentation. However, no quantitative results on skin detection were presented.

Ghazali *et al.* (2012) also used Gaussian skin-color model with Cg-Cr color space to detect skin color. The Gaussian model transforms the input image into a gray values image. Then, the gray values image is transformed into binary image using thresholding techniques where skin regions are set to 1's and background to zeros. Many other works such as (Cai & Goshtasby, 1999)

(Jebara, Russell, & Pentland, 1998) (McKenna *et al.*, 1998) (Oliver, Pentland, & Berard, 2000) also used Gaussian mixture models for skin detection.

3.7.4 Other Methods

Kim *et al.* (2008) proposed a skin color modeling approach in HSI color space, while considering intensity information by adopting the B-spline curve fitting to make a mathematical model for statistical characteristics of a color with respect to intensity. Li *et al.* (2007) proposed an algorithm based on facial saliency map. Juang and Shiu (2008) used self-organizing Takagi–Sugeno-type fuzzy network with support vector machine, which is applied to skin color segmentation. Gasparini and Schettini (2006) used Genetic Algorithm to find the classification boundaries between skin and non-skin pixels based on multiple color spaces. Gomez, Sanchez, and Sucar (2002) evaluated each color component for several color spaces, and then made a mixture color space from them for skin detection. The authors claimed that their approach can discriminate very well skin in both indoor and outdoor scenes.

Tan *et al.* (2012) proposed a human skin detection approach that consists of four steps. First, the system detects human faces from the source image. Second, based the detected face region, the system estimates the threshold value(s) for the skin-color using log opponent chromaticity (LO) color space. Third, the distribution of skin and non-skin colors is defined using 2D-histogram and Gaussian model. Finally, skin detection is done based on a fusion product rule of the two features. The system produced true positives and false negative of 65.80% and 34.20% respectively with general accuracy of 90.39%.

3.8 Region-based skin segmentation

Spatial information is useful as most segments corresponding to real world objects consist of pixels, which are spatially connected. The main idea of region-based segmentation techniques is to identify various regions in images that have similar features (Jain, 1989). The main region-based approaches can be categorized into:

- Region growing: The basic approach is to start the segmentation with seed pixels and from these, regions are grown by appending to each seed those neighboring pixels that have properties similar to the seed (such as same color) (Gonzalez *et al.*, 2007). In general, growing a region would be stopped if the surrounding pixels do not satisfy the conditions for inclusion in that region. Usually, seed pixels can be selected interactively by the user, or automatically using priori information about the nature of the problem.

  A problem with region growing is its inherent dependence on the selection of seed pixels (or region) and the order in which pixels and regions are examined (Cheng *et al.*, 2001). This means that the placement of initial seed points influences the result of region-based segmentation.

- Region splitting and merging: The idea in which the entire image is considered initially to be a single region. Certainly, the uniformity predicate will be false for this region, so it is divided into sub-regions. These sub-regions are then split and/or merged in an attempt to meet the uniformity criteria (Efford, 2000). The region merging is often combined with region growing or region splitting to merge the similar regions to make a homogeneous region as large as possible (Cheng *et al.*, 2001).

- Morphological Watersheds: The concept of watersheds is based on visualizing an image in three-dimensions: two spatial coordinates versus intensity levels. Rather than working on an image itself, this method is often applied on its gradient image. The gradient image is similar to topography with boundaries between regions. The segments correspond to the individual regions in the image. A gray-level image can be seen as a topographic relief, with the grey level of a pixel interpreted as its altitude in the relief. A drop of water falling on a topographic relief flows along a path to finally reach a local minimum. When we consider the image to be an altitude surface in which bright areas as high (correspond to ridge points), and dark areas as low (correspond to valley points), it is then natural to relate such surfaces in terms of catchment basins and watershed

lines. In such a topographic interpretation, three types of points are considered (Gonzalez & Woods, 2002):

a) Points belonging to a regional minimum.
b) Points at which a drop of water, if placed at the location of any of these points, would fall to a single minimum. These points are called *catchment basin* or *watershed* of that minimum.
c) Points at which water would be equally likely to fall to more than one such minimum. These points are called *divide lines* or *watershed lines*.

The ultimate goal is to find the *watershed lines* which can be considered as the region's boundaries. However, a main disadvantage of the watershed segmentation is that most of the time it leads to over-segmentation in the gradient method (Agathos, Pratikakis, Perantonis, Sapidis, & Azariadis, 2007). The method is generally based on complex concepts, requiring detail analysis and it can be computational expensive (Sonka *et al.*, 2008). Another disadvantage of watershed segmentation, related to the image noise and the image's discrete nature, is that the final boundaries of the segmented region lack smoothness. Hua, Abderrahim, Jaral, and Su (2003) stated that it is not an efficient idea to treat the watershed segmentation as the final segmentation.

Although region-based segmentation approaches are widely used in different image segmentation applications, the researchers found that it is rarely used for skin color detection as an initial step because they are computationally demanding and therefore time consuming.

To the best of our knowledge, the only work that used region-based approach for skin detection was done by Chen and Wang (2007); in which the authors proposed a two-staged skin detector. In the first stage, the detector applies complete region-based image segmentation based on color-texture to segment the source image into homogeneous regions. Then, in the second stage, a pixel-based segmentation is applied to extract the candidates "key skin region" through a set of rules in the RGB color space.

A summary of the skin detection approaches is shown in Table 3.1.

Table 3.1: Summary of skin detection approaches.

Year	Authors	Color space	Intensity comp.	Noise Removal	Data Analysis	Skin detection method	Pre-Training	Test Database	Diff. Skin Types	Diff. Illum.	TD	FP	FN
1997	Oliver *et al.*	RGB	NA	No	No	GMM	Yes	NA	Yes	Yes	NA	NA	NA
1997	Yang, Lu, & Waibel	Normalized RG	NA	No	No	SGM	Yes	NA	Yes	Yes	NA	NA	NA
1998	Yang M. and Ahuja	CIE-LUV	No	No	No	GMM	Yes	NA	Yes	No	NA	NA	NA
1998	Sobottka and Pitas	HSV	No	No	No	Thresholding	No	M2VTS	No	No	NA	NA	NA
1999	Garcia and Tziritas	HSV + YCbCr	No	No	No	Thresholding	No	100 Images	No	No	NA	NA	NA
1999	Peer and Solina	RGB	NA	No	No	Thresholding	No	M2VTS + PICS	No	No	NA	NA	NA
2000	Bergasa *et al.*	RGB	NA	No	No	SGM	Yes	NA	Yes	Yes	NA	NA	NA
2000	Oliver *et al.*	RGB	NA	No	No	GMM	No	NA	No	No	NA	NA	NA
2001	Brown	TSL	No	No	No	SOM	Yes	AP+IC	NA	NA	94.00%	NA	NA
2001	Cho and Jang	HSV	No	No	No	Adaptive Thresholding	Yes	379 Images	Yes	Yes	86.9-93.8%	NA	NA
2001	Greenspan *et al.*	Normalized RG	NA	No	No	GMM	No	ARH + ARL	Yes	Yes	NA	NA	NA
2001	Phung *et al.*	YCbCr	No	No	No	ANN	Yes	NA	NA	No	91.6	4.5	4.3
2002	Baskan *et al.*	HSV	No	No	No	Thresholding	No	AR	No	No	NA	NA	NA
2002	Hsu *et al.*	YCbCr	No	No	No	SGM	Yes	HHI	Yes	Yes	NA	NA	NA

Year	Authors	Color space	Intensity comp.	Noise Removal	Data Analysis	Skin detection method	Pre-Training	Test Database	Diff. Skin Types	Diff. Illum.	TD	FP	FN
2002	Jones and Rehg	RGB	NA	No	No	Histogram-Based	No	Compaq	Yes	Yes	80%	8.50%	NA
2002	Lee and Yoo	Multi	No	No	No	Elliptical model	Yes	Compaq	No	No	90.00%	35.70%	NA
2003	Kovac and Peer	YUV	No	No	No	Elliptical model	Yes	40+60 images	NA	Yes	NA	NA	NA
2003	Soriano *et al.*	Normed RG	NA	No	No	Adaptive thresh.	No	UOPB	Yes	Yes	NA	NA	NA
2003	Storring	RGB	NA	No	No	Thresh	No	UOPB	Yes	Yes	NA	NA	NA
2003	Seow *et al.*	RGB	NA	No	No	ANN	No	NA	NA	NA	NA	NA	NA
2004	Kakumanu	RGB	NA	No	No	Thresh	Yes	326 images	No	Yes	NA	NA	NA
2004	Sigal *et al.*	HSV	Yes	No	No	Bayes	Yes	Compaq	Yes	Yes	86.84%	NA	NA
2004	Sebe *et, al.*	RG	NA	No	No	Bayesian Network	No	Compaq	No	No	87.66 - 98.32%	1 - 10%	NA
2005	Kim *et al.*	HIS	No	No	No	Fuzzy Rules	Yes	NA	No	Yes	NA	NA	NA
2005	Zaquot *et al.*	RGB	NA	No	No	LUT	No	PICS	No	No	94.17	17.31	NA
2005	Liu *et al.*	YUV	No	No	No	SGM	No	NA	No	No	NA	NA	NA
2007	Chen and Wang	RBG	NA	No	No	Region-based +Thresholding	No	3000	Yes	Yes	92.67%	6.17%	NA
2007	Do *et al.*	HSV	Yes	No	No	Thresholding	No	PBFD	No	Yes	82.70%	27.40%	17.30%
2008	Shih	YCbCr	No	No	No	GMM	No	NA	No	No	NA	NA	NA

Year	Authors	Color space	Intensity comp.	Noise Removal	Data Analysis	Skin detection method	Pre-Training	Test Database	Diff. Skin Types	Diff. Illum.	TD	FP	FN
2008	Vadakkepat	YUV + YCbCr	No	No	No	Thresholding	No	NA	Yes	No	NA	NA	NA
2009	Han	YCbCr	No	No	No	SVM	Yes	ECHO	Yes	No	86.34%	0.96%	NA
2010	Moallem *et al.*	HSI	No	No	No	Fuzzy Rules	Yes	HHI + Champion + IMM + Essex + Bao + Caltech	No	Yes	NA	NA	NA
2010	Taqa and Jalab	RGB	NA	No	No	ANN	Yes	NA	No	No	95.61%	0.87%	NA
2011	Yuetao	YCbCr+RGB	No	No	No	SGM + Thresholding	Yes	100 Images	No	No	NA	NA	NA
2012	Fernandez 2012	TSL	No	No	No	Histogram-Based	No	NA	No	No	90%	NA	NA
2012	Ghazali *et al.*	YCbCr	No	No	No	SGM	Yes	NA	No	No	NA	NA	NA
2012	Tan *et al.*	LO	No	No	No	SGM	No	ETHZ PASCAL + Stottinger + Pratheepan	Yes	No	65.80%	5.77%	34.20%

3.9 Summary

Recently, low cost color sensors and other hardware equipment for processing color images have become available everywhere. New computers (e.g. PCs and laptops) are sold already bundled with colored digital cameras and video cameras. Color in general provides additional information that we may be able to exploit to improve image segmentation and object detection. Color is a property of great significance to human visual perception. However, historically it was not particularly used in digital image processing until 1980s (Sonka *et al.*, 2008). Image processing techniques of color images are now needed in a broad range of applications. One of the segmentation tasks in image analysis is to find regions of specific color in a given color image.

Skin color detection has been proven to be a cue feature for face detection, localization, and tracking, as it has the advantages of being orientation and scale invariant, and facilitates low computational cost (Do *et al.*, 2007).

Detecting skin-colored pixels, although seems a straightforward easy task, has proven to be quite a challenging task in complex images that are captured under unconstrained imaging conditions (Kakumanu *et al.*, 2007).

Currently, although there are many skin color models in the literature, there is a limitation about how to measure the correctness of a model and to what range is the validity of this model (Gonzalez *et al.*, 2007; Russ, 2007). In general, the performance of a model depends on many factors, such as the color space that is used, the shape of the distribution, the parameters used, nature of data, size of samples for training, image characteristics, noise data, etc.

The skin-color modelling methods can be classified into three categories: Explicitly defined thresholding, nonparametric, and parametric methods. In general we can show the general characteristics of skin-color modeling methods:

- **Explicit Defined Skin Color Thresholding Methods (Classification rules):**
  - The simplicity of the classification rules.
  - Easy to adjust.
  - It is computationally inexpensive.
  - The correctness of the model depends on the thresholding values or classification rules.
  - Machine learning algorithms can be used.
  - Sometimes, it is difficult to find the thresholding values to describe the actual distribution.

- **Non-parametric methods**
  - Using Look-up table method makes the system very fast compared to other methods such as Distance-based, NN, Fuzzy logic.
  - Independence of distribution shape.
  - Requires much representative training dataset

- **Parametric methods**
  - Does not require much representative training dataset.
  - The correctness of the model depends on the distribution shape.
  - It is computationally expensive during training phase. For example, as reported by (Jones & Rehg, 2002), the mixture of Gaussians models took about 24 hours to train both skin and non-skin models using 10 Alpha workstations in parallel. In contrast, the histogram models could be constructed in a matter of minutes on a single workstation.

- It is computationally expensive during classification phase since all of the Gaussians must be evaluated in computing the probability, mapping, and then thresholding.
- Many works such as those by (Jones & Rehg, 2002) and (J. Y. Lee & Yoo, 2002) show the limitations of the single and the mixture of Gaussians models; describing that other methods can be superior in accuracy.
- From the standpoint of storage space, however, the mixture model is a much more compact representation of the data.

- **Region-based methods**
  - This type of methods are highly computational demanding and therefore time consuming.
  - Many challenging factors such as illumination variations, noise, and camera characteristics may lead to inaccurate segmented regions.

CHAPTER FOUR

METHODOLOGY OF SKIN COLOR MODELING AND DETECTION

4.1 Introduction

In this research work, the use of skin color feature for image segmentation is motivated by two main factors compared to other features. First, skin color is a powerful feature that often simplifies the detection and extraction of human targets from a scene with low computational cost. Second, color is robust against object rotation, scaling, and partial occlusion. However, segmentation of complex image(s) is a challenging task as colors in images are impacted by various factors, such as illumination, different ethnic groups, complex background, camera characteristic, etc. Although many skin detection solutions are proposed in literature to cope with these challenges (see Chapter 3), these solutions still suffer from the following. ***1) Low accuracy*:** most skin detection methods show high False Negative and/or False Positive errors when dealing with complex images (i.e. images captured under unconstrained imaging conditions). ***2) High computational cost:*** many existing methods are un-preferable to be applied in real-time face processing systems due to the high computational cost.

This chapter presents a reliable skin color modelling and detection approach. The proposed approach can overcome sensitivity to variations in lighting conditions, ethnic groups, complex backgrounds, and camera characteristics. To the best of our knowledge, this is the first attempt that employs multi-skin models for the skin detection problem. Furthermore, the proposed approach has the advantage of using a Lookup Table for the pixel-based skin segmentation rather than computations, which makes it very fast.

Additionally, a novel method for testing and evaluating image segmentation methods in application to skin detection problem is proposed in this chapter. Furthermore, we have

provided detailed experiments of such evaluation. We believe that this work is the most comprehensive and detailed exploration of skin color modeling.

The remainder of this chapter is organized as follows. Data collection is described in Section 4.2. Choosing the suitable color space is described in section 4.3. Design issues are discussed in Section 4.4. Estimating the skin color space is described in Sections 4.5. Building skin-color models is described in Section 4.6. Pixel-based segmentation and region-based segmentation are described in Sections 4.7 and 4.8 respectively. Skin-color modeling is described in Section 4.9. A novel procedure for testing and evaluation skin segmentation methods procedure is described in Section 4.10. The proposed algorithm for building skin models is described in Section 4.11. Comparison with other works is presented in Section 4.12. The applicability of this approach for other applications and chapter's summery are described in Sections 4.13 and 4.14 respectively.

4.2 Data Collection

In this research, the Microsoft Picture Manager and Adobe Photoshop CS3 packages were used to cut patches of skin and non-skin samples from real images manually. Our dataset of patches is composed of 24,328,670 pixels collected from three public face databases plus our own face database. These are FEI, CVL, LFW, and FSKTM.

- **The FEI face database** is a Brazilian face database that contains a set of face images taken at the Artificial Intelligence Laboratory of FEI university in São Bernardo do Campo, São Paulo, Brazil. This database includes 200 persons × 14 images for each person, a total of 2800 images. All images are colorful and taken against a whitish homogenous background in an upright frontal position with profile rotation of up to about 180°. Scale might vary about 10% and the original size of each image is 640×480 pixels. All faces are mainly represented by students and staff at FEI, between 19 and 40 years old. The number of male and female subjects is exactly the same and equal to 100. Figure 4.1 shows some examples of image variations from the FEI face database.

Figure 4.1: Samples of FEI face Database.

- **The CVL Face Database** of the Computer Vision Laboratory in the University of Ljubljana, Slovenia. This database includes 114 persons × 7 images for each person. Each image is 640×480 pixels. Images were taken under different lighting directions, expressions, and poses. Examples of image variations from the CVL face database are shown in Figure 4.2.

Figure 4.2: Samples of CVL face database.

- **The LFW Face database**, Labeled Faces in the Wild (including the **FDDB** dataset), from the University of Massachusetts, Amherst, USA. This database is designed for studying the problem of unconstrained face recognition. It contains more than 13,000 face images collected from different resources with different imaging conditions. The database represents an initial attempt to provide a set of labeled face photographs spanning the range of conditions typically encountered by people in their everyday lives. The database exhibits "natural" variability in pose, lighting, focus, resolution, facial expression, age, gender, race,

accessories, make-up, occlusions, background, and photographic quality. Despite this variability, the images in the database are presented in a simple and consistent format for maximum ease of use. The Face Detection Data Set and Benchmark (FDDB) dataset, which is part of LFW, is designed for studying the problem of unconstrained face detection. This database contains 2845 images with 5171 faces. A few examples of the LFW dataset are shown in Figure 4.3.

Figure 4.3: Samples of LFW face Database.

- **FSKTM face database (Our Database).** This database consists of 540 images collected from different sources and includes various types. We noticed that the above-mentioned databases (or other public databases) are single face images and designed mainly for face recognition purpose (i.e. usually contain mugshot-style images on uniform backgrounds). The main reason for constructing our own database is to collect multi-face images belonging to different ethnic groups. In general, Internet images are acquired under unknown conditions. These images, therefore, have not been taken by the same camera and of course taken under various lighting conditions. Many students and staff in the Faculty of Computer Science and Information Technology (FSKTM) at University of Malaya have digital images online. These images were also used. A few examples of our dataset are shown in Figure 4.4.

Figure 4.4: Samples of FSKTM face database.

In this research, we do our best to include the most important random factors that can impact skin color appearance (or color-tones) in image(s):

- **Different races:** Samples are collected from people belonging to several ethnic groups such as European, African, and Asian origins.
- **Lighting conditions:** Samples are collected from real images with different lighting conditions such as uniform lighting, non-uniform lighting, indoor and outdoor images.
- **Lighting reflections and shadows:** Skin pixels containing strong highlights, medium shadows, and dark shadows are included.
- **Different parts of human body:** Samples are collected from several regions of the human body including the cheek, forehead, nose bridge, shoulders, arms and legs in order to cover different skin tones.
- **Camera characteristics:** If the skin samples are collected from images of a specific face database that uses a specific camera, then the results depend mainly on testing images using the same camera. When using test images of another database, segmentation system will generally fail. Therefore, to seek for robustness of our system the samples are collected from real images of different sources and types (i.e. four face databases) to make sure that the skin samples were taken by different cameras.

Figure 4.5 shows samples of the skin and non-skin patches (i.e. training data) collected from the above-mentioned databases. Figure 4.5(a) shows skin samples collected from whitened skin people. Figure 4.5(b) shows skin samples collected from reddish skin regions. Figure 4.5(c) shows skin samples collected from blackish skin people. Figure 4.5(d) shows non-skin samples collected from background. We also collected samples of lighted skin regions which are not included in this figure (i.e. because these patches tend to white color and when printed on white paper, the viewer can hardly recognize these patches).

In general, collecting training data implies many issues and sub-problems such as noise data, data generalization, and overlapping between classes. These issues will be discussed in detail in Section 4.11.1.

Images containing strong colors which tend to be unreal colors are discarded, such as images which are completely tend to blue, violet, etc. Color correction and color enhancement of the images containing strong colors which tend to be unreal are beyond our goals. The possible reasons for mapping real normal color to false-color images have been discussed in Section 3.4.

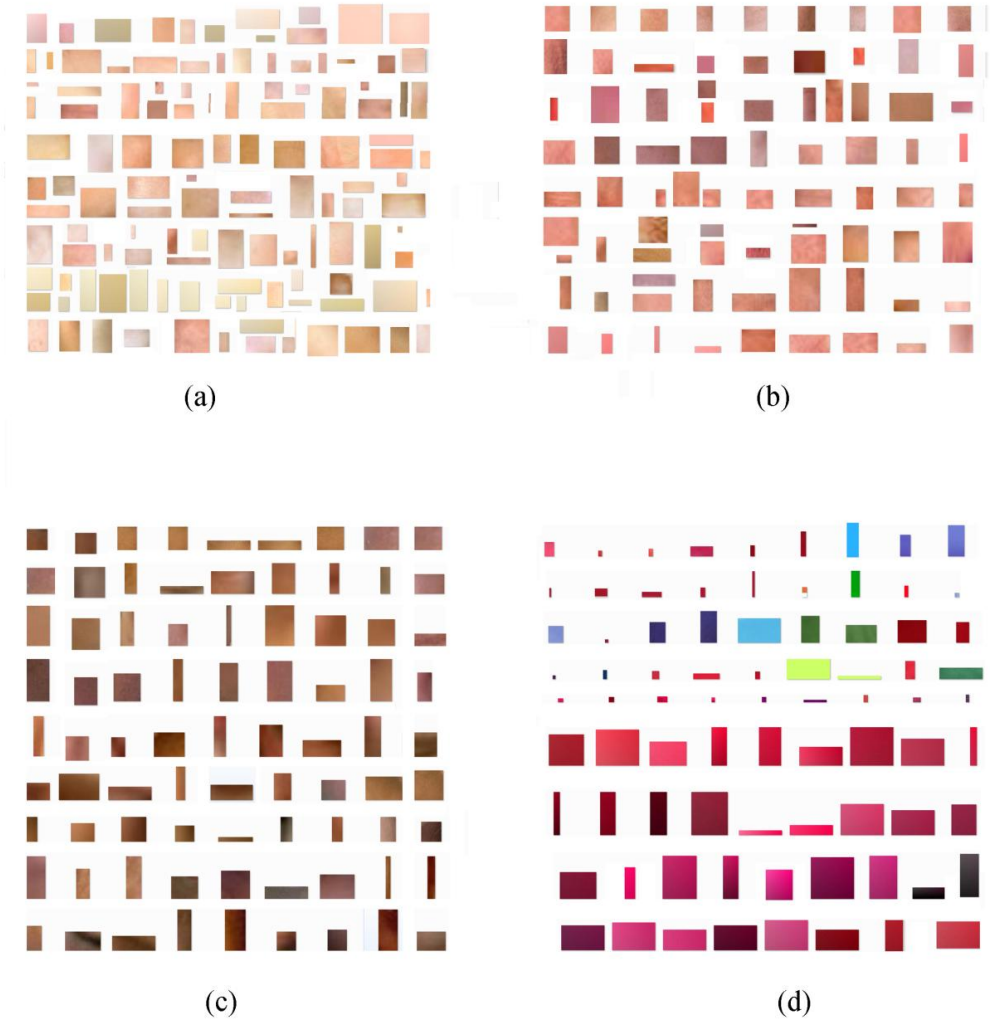


Figure 4.5: Skin and non-skin samples (training data); (a) skin samples collected from whitened skin people; (b) skin samples collected from reddish skin regions; (c) skin samples collected from blackish people; (d) non-skin samples collected from background.

4.3 Choosing the Suitable Color Space

In this research work, the HSV color space was used for skin-color modeling and image segmentation for the following reasons:

1) It is well known that the robustness of human skin detection approach against variations in illumination and ethnicity can be accomplished if the color space efficiently separates the chrominance from the luminance information (Chen & Wang, 2007; Kovac *et al.*, 2003; Srisuk & Kurutach, 2002; Terrillon *et al.*, 1998). The HSV model has this property and can be adopted to solve the illumination variation problem. Thus, by using the HSV color space we try to improve the final results of skin detection.

2) Compared to other color models, the HSV color space tends to be more realistic in that its representation strongly relates to the human perception of color (Gonzalez et al., 2007). The RGB, YUV, CMY, etc., color spaces are suitable for technical aspects. They do not correspond to the way that people recognize or describe colors. As mentioned before, one neither refers to the color of a car by giving the percentage of mixing the three primary colors, red, green, and blue, nor the percentage of mixing three pigments, cyan, magenta, and yellow. Usually, humans tend to describe the color of an object by its hue, saturation, and intensity such as blue, dark blue, etc.
3) The HSV model is an ideal tool for developing image processing algorithms based on color descriptions (Efford, 2000; Russ, 2007; Sonka *et al.*, 2008). To illustrate this point, an interesting example illustrated by Burdick (1997) shows the effectiveness of HSV model compared to other models - suppose we might want to change the color of a bright yellow car to blue, but we want to leave the rest of the scene, including highlights and shadows on the car, unaffected. This would be an impossible task in RGB, but it is relatively simple in HSI or HSV, because the yellow pixels of the car have a specific range of Hue, regardless of intensity or saturation. Therefore, those pixels can be isolated and their Hue components modified, thus giving a different color car leaving shadows on the car unaffected.
4) According to Chaves-González *et al.* (2010) and Zarit *et al.* (1999), the skin color cluster is more compact in HSV. The authors reported that HSV color space achieves the best results for skin color detection compared to other color spaces.
5) The classification boundaries are more easily adjusted using the HSV model, because of a direct access to its color components (Garcia & Tziritas, 1999).

Another perceptually uniform color space is the HSI color space, which can be used for this purpose. Although they are so similar, the HSV was chosen for two reasons. First, we had more experience with HSV. Second, our system was implemented using MATLAB Software (2010a) in which the Image Processing Toolbox (IPT) provides a set of built-in functions to transform RGB data to three common color spaces, and vice versa. These are HSV, YCbCr, and YIQ only. MATLAB software supports HSV over HSI.

4.4 Design Issues of Skin Color Modeling and Detection

In this section, four design issues are presented concerning the cost of False Negative/False Positive errors, dimensionality of color space, color quantization, and simplicity.

4.4.1 False Negative (FN) and False Positive (FP) Costs

Skin color segmentation aims at building a skin color model or classification boundaries to discriminate between different classes (i.e. skin and non-skin pixels). We sought to use an algorithm (or classifier) that would take the measured features of an unknown pixel (i.e., three-color components) as input and then predict the true class membership as output. The term "predict" indicates that this process may not always be possible without error. The existing skin segmentation methods may cause two kinds of classification errors: False Negative (FN) error in which a skin pixel is classified as a non-skin pixel, and False Positive (FP) error in which an image pixel is classified as skin pixel, although it is not (Zainuddin, Naji, & Al-Jaafar, 2010).

In general, complex backgrounds usually increase False Positive (FP) errors due to the fact that natural scenes contain many objects with skin-like colors. On the other hand, variations in illumination, ethnic groups, and camera characteristics usually increase False Negative (FN) errors.

We should realize that classification errors are rarely avoided even for an ideal application. Moreover, it is generally accepted that each type of classification error has an associated *cost* or *risk*. It is possible to assume that the consequences of classification errors are equally costly. For example, classifying a novel pixel as a non-skin when in fact it is skin, is just like the cost of the converse. Such a symmetry in the *cost* is often in many systems, but not always true for all systems.

In this research, along with the most important point of the requirements of the proposed system (that is: all the true faces should be detected), the costs of the segmentation errors FP and FN are

not equal. The most critical problem is the FN rate, attributed to the fact that image segmentation is the first step in image analysis. When a face is missed, the post-processing stages of the system cannot get it back. Therefore, the overall success or failure of the whole system depends mainly on this important issue. In contrast, FPs can be eliminated by the system's subsequent steps. It is expected that the detected skin-tone regions will include some non-face regions whose color is similar to skin-tone. The post-processing steps such as facial features detection procedure are used to reject 'skin' regions that do not contain any facial features. Therefore, researchers often design their classifiers in such a way that it minimizes the expected cost (Duda *et al.*, 2001). So, our true task in this research is to minimize the FN rate.

As such the idea in this research is different from that of other works. We insist that the output of image segmentation must include all objects that have skin like color. In other words, objects which are seemingly similar to skin colors should be treated as objects of interests, and should be retained in the image segmentation process.

4.4.2 Dimensionality of Color Space

Modeling skin color requires choosing an appropriate color space and identifying a cluster associated with skin color. In real world cases with diversity of image types and sources, accurate skin color clustering model becomes a difficult task. Variation in skin color among different racial groups, lighting conditions, and camera characteristics affect the appearance of skin color and hence the performance of skin model. As shown in Section 3.6, most color spaces (e.g. RGB, HSV, YCbCr, CYMY) represent color as tuples of numbers, typically as three or four values called color components or channels. Russ (2007) stated that for a three-dimensional color space there is no easy or obvious way with present display or control facilities to interactively enclose an arbitrary region in three-dimensional space and see which pixels are selected, or to adjust that region and see the effect on the image

Unfortunately, several previous works argued that although different people have different skin color, the major difference lies largely between their intensity rather than their chrominance (Juang & Shiu, 2008; Kumar & Bindu, 2006; Shih *et al.*, 2008; Srisuk, Kurutach, & Limpitikeat, 2001). They assumed that the chrominance components of the skin-tone color are independent of the luminance component. Consequently, the illumination channel is placed in the non-useful zone and a two-dimensional color space is chosen instead of a three-dimensional color space to ease the determination process of the skin color clustering model. This can be summarized as follows:

- The HS replaces the HSV color space (Baskan *et al.*, 2002; Juang & Shiu, 2008; McKenna *et al.*, 1998; Sandeep & Rajagopalan, 2002; Sobottka & Pitas, 1998; Tsekeridou & Pitas, 1998).
- The CbCr replaces YCbCr color space (Chai & Ngan, 1999; Ghazali *et al.*, 2012; Habili *et al.*, 2004; Kumar & Bindu, 2006; Shih *et al.*, 2008; Yuetao & Nana, 2011).
- The YI replaces YIQ (Wei & Sethi, 2000).
- TS replaces TSL color space (Tomaz *et al.*, 2004).
- UV replaces CIE LUV (M. H. Yang & Ahuja, 1998).

Unless some pre-assumptions are imposed (e.g. uniform lighting), such approaches show high false detection rates due to loss of some color information when an image is expressed in a low-dimensional space instead of a high-dimensional space. Simply ignoring any piece of color information affects the system accuracy (Moallem et al., 2011).

In this research work we decided to use the full color information for building our skin-color models to achieve the best image segmentation results.

4.4.3 Color Quantization

Complex color images contain millions of distinct colors. For example, HSV color space (that uses 24-bits to represent colors) offers about 16.7 million distinct colors. For any color model or segmentation approach, one of the goals for effective and efficient computational cost is to reduce the number of colors used to represent the contents of an image. This process is the color *quantization*, i.e. each set of points of similar color is represented by a single color.

In this research, the quantization is done at Hue channel (or Hue wheel; where $0° \leq \text{Hue} \leq 360°$). The Hue wheel is divided into equal intervals of 6 degrees. Thus, the colors (or hue) in the color space are reduced to 60 primary colors only (i.e. 360/6=60). From our point of view, this set of 60 colors is exhaustive enough to preserve the look of objects, taking into account the scale of what human eyes can differentiate between colors on close scrutiny of the hue in the color space. We strive to balance between the computational cost and the exhaustive range of colors.

Once the number of colors has been sufficiently reduced, the graphical representation can be done easily. The 3D space is transformed to 2D subspace by creating SV-subspace for each quantized hue. Figure 4.6(a) shows the standard representation of HSV color space. Figure 4.6(b) shows the HSV cylinder representation. Figure 4.6(c) shows SV-plane for a known hue H=06°. Therefore, we have 60 slides of SV-subspaces. For each slide, *H* is constant while saturation *S* and value *V* are variables representing the x-axis and y-axis respectively. The plane is displayed graphically where S and *V* are in range [0,100].

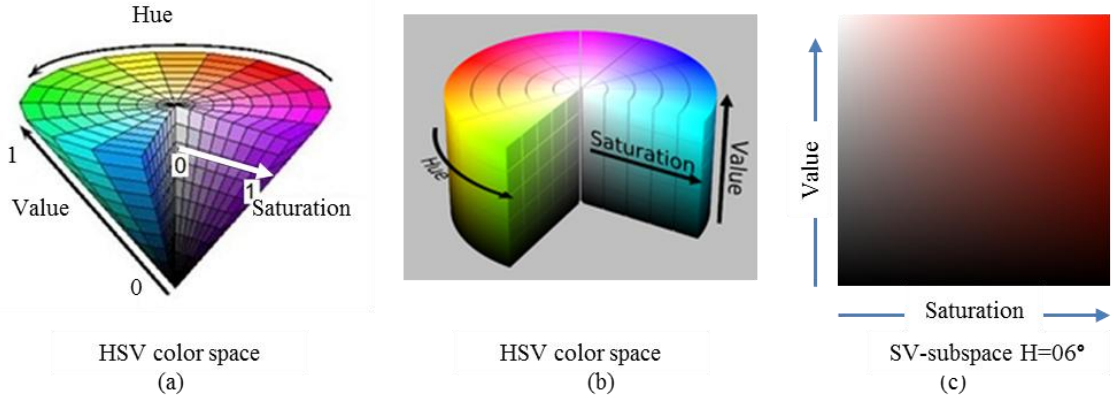


Figure 4.6: Color quantization of HSV color space. (a) HSV color space cone representation; (b) HSV cylinder representation; (c) 2D SV-subspace for Hue=06° while Saturation *S* and Value *V* are variables representing the x-axis and y-axis respectively.

4.4.4 Simplicity

The simplicity is related to the classification process in terms of low computational cost.

4.5 Estimating the Skin Color Space

The HSV color space represents color as tuples of numbers called color components or channels. The color space represents a coordinate system where each specific color is represented by a single point in the coordinate system. As mentioned before, the training data (or pixels) were collected manually from different datasets (Section 4.2). Based on our training data, Figure 4.7 shows the distribution of the skin-color samples in 3D HSV model where Hue component is in range [-180°, 180°] circular, Value *V* and Saturation *S* are in range [1,100]. The figure shows that human skin color forms a cluster in color space. Unfortunately, there is no easy or obvious way with present display facilities to interactively enclose an arbitrary region in three-dimensional space and see which pixels are selected, or to adjust that region and see the effect on the image (Russ, 2007). By considering only one channel that is Hue (or color); the frequency of skin color tones at Hue channel is shown in Figure 4.8. The Figure 4.8(a) shows skin color distribution at Hue channel where Hue is in range [0°, 360°]. The Figure 4.8(b) shows the distribution where Hue is in range [-180°, 180°] circular. We found that the maximum frequency of human skin is at Hue=18°.

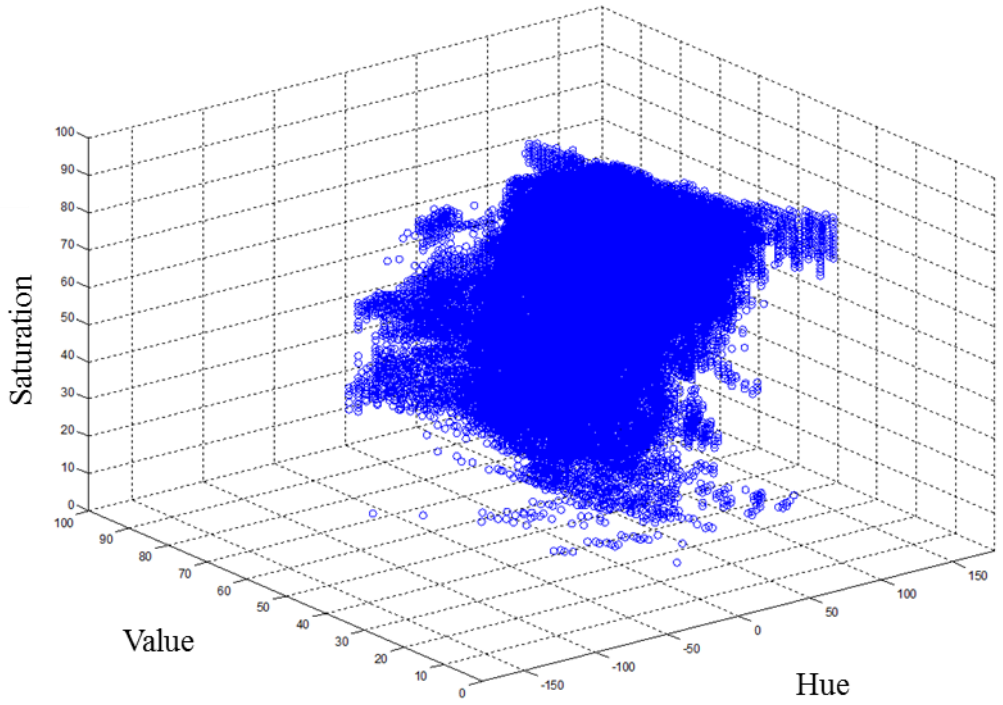


Figure 4.7: Skin samples distribution in 3D HSV model where Hue= -180° to 180° (circular).

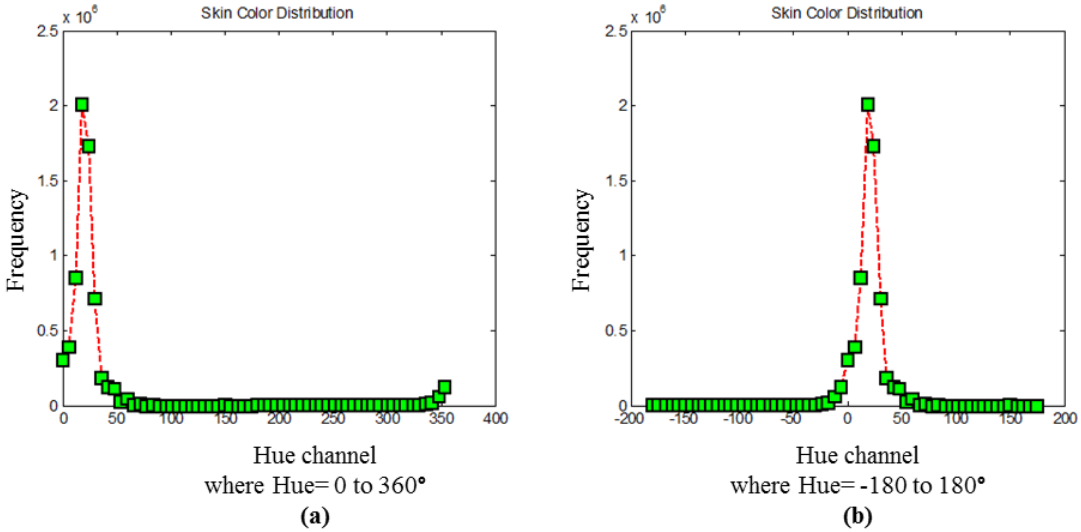


Figure 4.8: Frequency of human skin color at Hue channel. The maximum frequency is at Hue=18°; (a) Hue=0° to 360°; (b) Hue= -180° to 180° (circular).

Based on our training data shown in Figure 4.8 and from the feedback of skin detection experiments through this research, we found that the hue of skin color clustering is in range [330°, 60°] as shown in Figure 4.9(b) where that standard representation of HSV color space is shown in Figure 4.9(a).

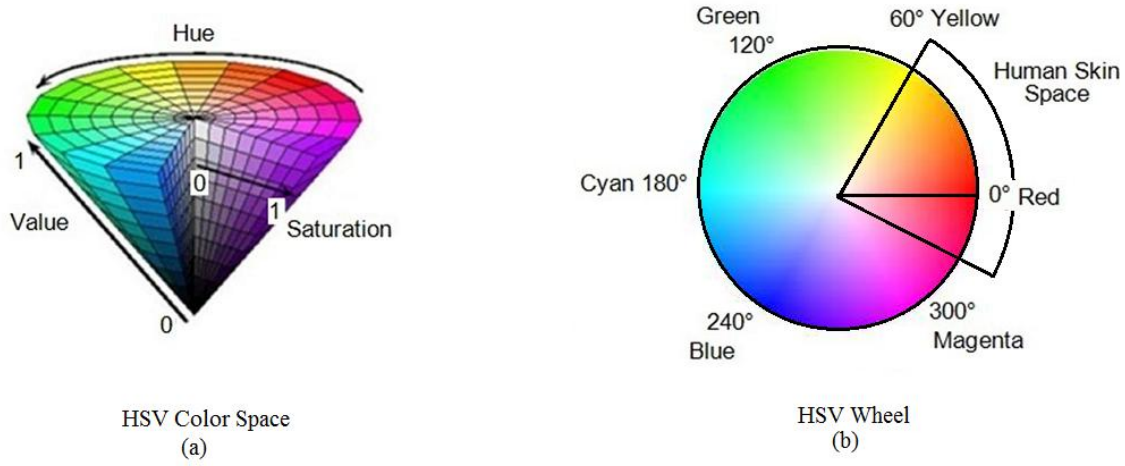


Figure 4.9: Skin-color space using HSV color space; (a) HSV color space; (b) HSV wheel identifying skin-color space.

4.6 Multi-Skin Color Models

When developing a system in which image segmentation is an integral component, the choice of color modeling method can directly affect the performance of the system as a whole. The challenging problem is how to select a skin color model that suits skin pixel classifications in complex images under unconstrained imaging conditions.

Before we built our skin color models, initially the research follow these steps. ***1)*** Three different skin-color modeling and detection were used based on three methods adopted from Baskan *et al.*(2002), Solina et al. (2002), and Chen & Wang (2007). ***2)*** Extensive skin detection experiments were conducted to identify FN/FP errors. Real images from the above-mentioned datasets are used for this purpose. ***3)*** We identified, compared, and analyzed the FN/FP errors of these methods to determine why these FNs/FPs make the problem difficult (some skin detection experiments are discussed in Section 4.10).

Based on the above-mentioned steps, we found two main reasons behind these cases. The first lies in the limitations of a single skin clustering model to cover different skin color tones, such as dark shadow regions and blackish skin. Strong light reflection may cause skin color information to be lost. In addition, makeup, montage, and image reproduction influence the skin color appearance to a reddish concentrated appearance. Thus, if the skin-color clustering model is too general, it may yield a large number of FP errors, that is, a non-skin pixel classified as a skin pixel. On the other hand, if the skin-model is tight or too specific, then it may yield numerous FN errors, in which the skin pixels are missed.

The second main reason is due to the fact that each colored pixel is treated individually in relation to the color space (skin or non-skin pixel), without considering the content of neighboring pixels.

Although many previous works assumed there was no relationship between the chrominance components and luminance components of the skin color (i.e. as mentioned in Section 4.4.2), our experiments using HSV color space show that the skin color differs in both luminance and chrominance. The experiments based on our training data show that the distributions of skin color for different ethnic origins are clustered separately. Figure 4.10 shows the distribution of skin color for two sets of skin samples (i.e. European origins and African origins) using HSV color space. The standard SV-plane at Hue=24 is shown in Figure 4.10(a). Figure 4.10(b) shows our training data distribution of normal white skin samples (e.g. European origins). Figure 4.10(c) shows our training data distribution of skin samples belonging to individuals of dark skins (e.g. African origin). Although the hue *H* is the same, the figure shows that white skins are clustering about a representation with low saturation and high intensity, while dark skins (e.g. African origins) are clustering about a representation with high saturation and low intensity. If both intensity and saturation are high (or both are low) then the color tends to be non-skin (e.g. the upper-right UR corner and lower-left LL corner of SV-plane) although the hue component is within the skin color space (i.e. in range [330°, 60°]). Figure 4.11 shows the maximum frequencies (i.e. peaks) using SV-plane. Figure 4.11(a) shows the maximum

frequencies based on row-column. Figure 4.11(b) shows the maximum frequencies based on columns. Figure 4.11(c) shows maximum frequencies based on rows. As shown in these figures, the upper-right UR corner and lower-left LL corner of SV-plane are evidently tend to be non-skin.

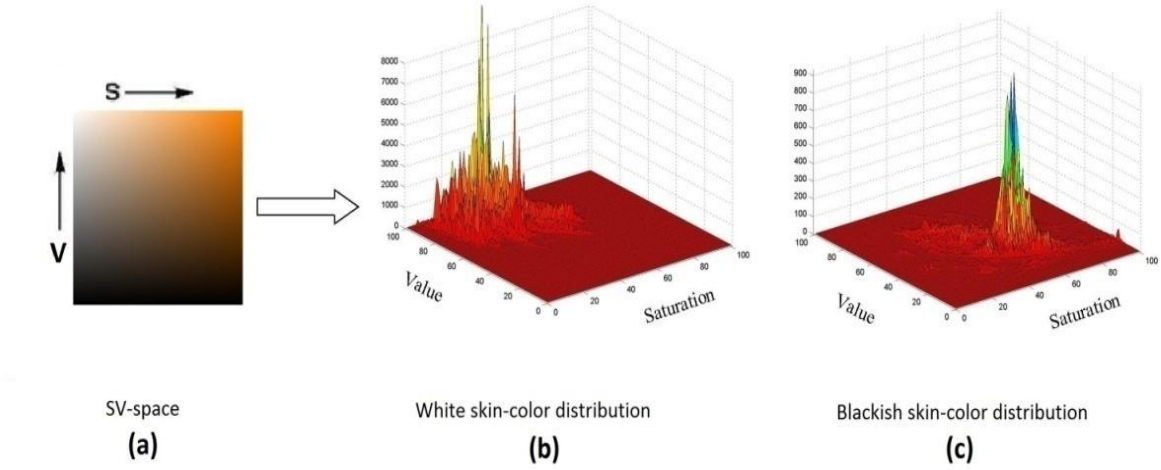
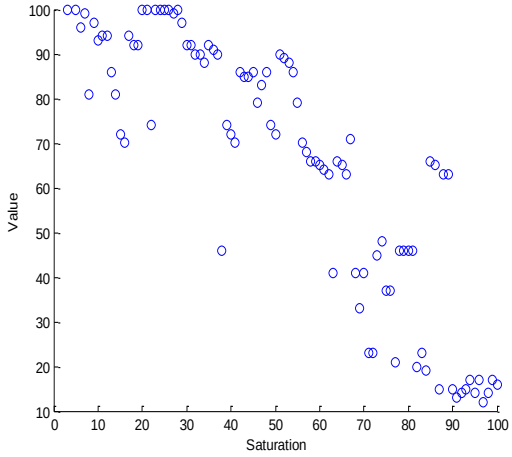
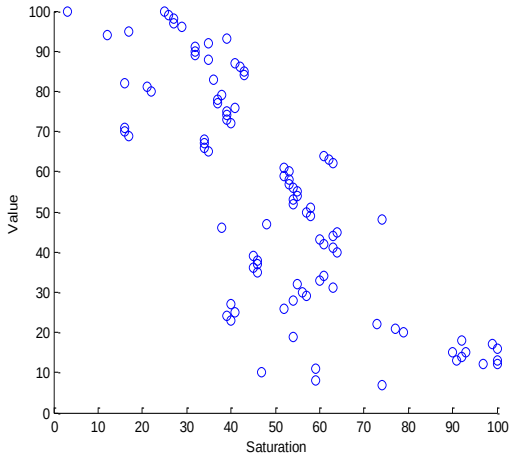


Figure 4.10: Skin-color distribution; (a) SV-space where hue=24; (b) skin-color distribution of skin samples belonging to white-skinned people (e.g. of European origin); (c) skin-color distribution of skin samples belonging to dark-skinned people (e.g. of African origin).

(a)



(b)



(c)

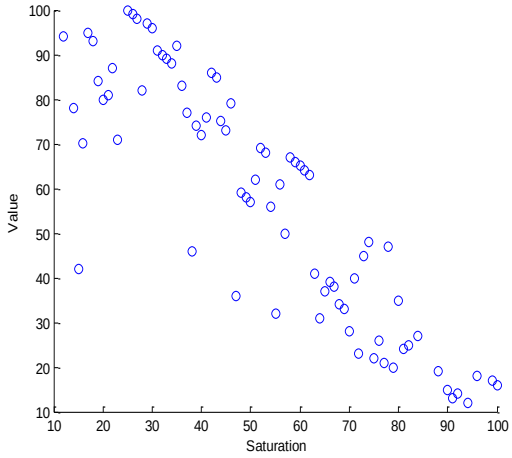


Figure 4.11: Maximum frequencies of skin samples; (a) maximum frequencies based on row-column; (b) maximum frequencies based on columns; (c) maximum frequencies based on rows.

Therefore, we found that the relationship between intensity *V* and saturation *S* components is "Inverse Relationship" in the SV-space. As the saturation *S* increases, the intensity *V* decreases, while, as saturation decreases the intensity *V* increases. So, we disagree with many previous

works such as (Baskan *et al.*, 2002; Chai & Ngan, 1999; Ghazali *et al.*, 2012; Habili *et al.*, 2004; Juang & Shiu, 2008; Kumar & Bindu, 2006; McKenna *et al.*, 1998; Sandeep & Rajagopalan, 2002; Shih *et al.*, 2008; Sobottka & Pitas, 1998; Srisuk *et al.*, 2001; Tomaz *et al.*, 2004; Tsekeridou & Pitas, 1998; Wei & Sethi, 2000) that argued and assumed there is no relationship between the chrominance components and luminance components (i.e. they are independent).

After we investigated experimentally the results and found that the distributions of skin color for different ethnic origins are clustered separately (Figure 4.10), the skin-color clustering problem indicates a necessity to think of a novel approach for human skin color segmentation that overcomes the problem of different skin color tones. Our approach suggests building multi-skin color clustering models, instead of a single skin model, whereby each model represents a cluster of a skin tone. The skin color models derived are:

m_1 : Standard white skin.

m_2 : Shadow skin (or blackish skin).

m_3 : Reddish concentrated skin and lips.

m_4 : Light-colored skin.

Furthermore, since non-skin samples are clustered in two disjoint regions, we decided to treat them as two different classes (rather than one) to get finest classification boundaries. Accordingly, the non-skin models derived are:

m_5 : Non-skin UR (upper-right corner)

m_6 : Non-skin LL (lower-left corner)

With our training data, the distribution of the six above-mentioned classes is shown in Figure 4.12. The figure shows high overlapping between classes due to many factors (see Section 4.11.1). The white regions in this figure are colors not encountered in any of the training data (i.e. missing data).

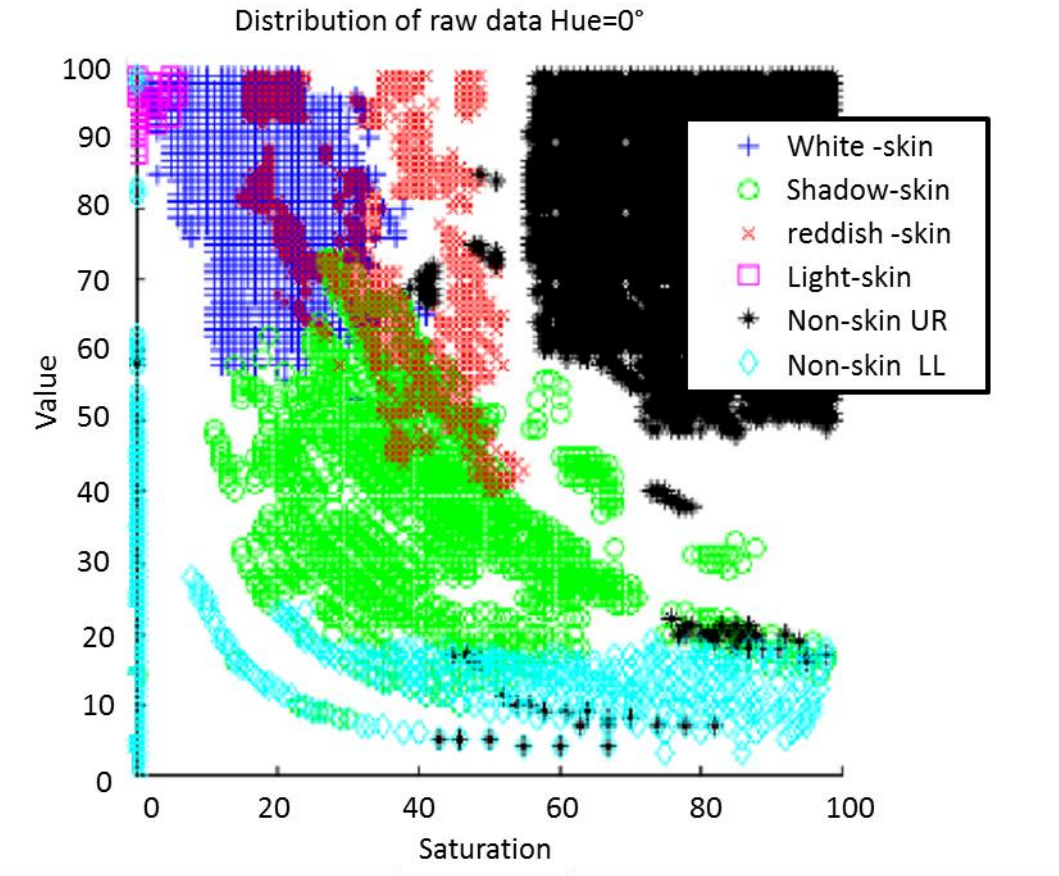


Figure 4.12: Distribution of our raw training data, Hue=0°.

This novel approach is completely different from that of other works. To the best of our knowledge, all previous works treated the collected data set of skin samples as one class. They formulated the problem as a two-class classification problem: skin and non-skin. This means that they treated all skin tones as one model. The main drawbacks of uni-skin modeling approach are:

- Low accuracy due to the limitation of one skin color model to cover various skin-color tones.
- The output of image segmentation based on uni-skin modeling would be a single binary image identifying the skin regions. With one binary image (or one layer), we do not have additional information about skin regions. Therefore, we cannot answer the question: "What are the properties of a specific skin region?" due to the fact that the output of skin segmentation, even for different ethnic individuals, would be identified in only one layer.

The key contribution of our method is that it is the first attempt that is based on multi-skin modeling in such a way that the classification boundaries contain information about the skin color of various ethnic origins and illuminations. The main advantages of the proposed method can be summarized as follows:

- This method overcomes the limitations of a uni-skin modeling which makes it capable to detect skin regions with higher accuracy. When classification boundaries contain information about the skin color of various races and illumination, the probability of missing faces would be reduced. In other words, pixels which are indistinguishable in one model may be fully distinguishable in the other model.
- To exploit more information about skin regions and the relationship between these regions. This is an important issue in image segmentation.
- This method of image segmentation takes into consideration the other steps of the system. Many previous works treated skin segmentation step as a standalone step and consequently ignored the link of this step with the other steps of the system. For instance, automatic illumination correction (or color correction) is an important step that is highly related to image segmentation. When we consider this issue in image segmentation methodology, the way to build skin color models might change accordingly. Our conception is that image segmentation and skin color correction (illumination correction) are so closely related that they should not be performed separately. When developing a system in which image segmentation is an integral component, the choice of skin color modeling method used can directly affect the subsequent steps as a whole.

The step-by-step algorithm used for finding the classification boundaries of these models is described in Section 4.11.2. Considering this, the classification boundaries (or decision functions) of these models are transformed into three-dimensional Lookup-Table (3D-LUT) to be used by skin detector program. A Lookup-Table is a data structure, that stores numeric data in a multidimensional array format, used to replace a runtime computation with a simpler array

indexing operation. In this research, we refer to this lookup-table as SD-LUT. The properties of the SD-LUT are:

i) It is constructed and pre-calculated offline.

ii) It is indexed by a color information vector (H, S, and V).

iii) Each SD-LUT cell contains the classification result of an indexed color.

iv) It is stored in the system secondary storage.

When the skin detector program is initialized, it just reloads the SD-LUT from the secondary storage. Since retrieving a value from a Lookup-Table is often faster than undergoing 'expensive' calculations at run time, the savings in processing time makes our method very fast.

4.7 Pixel-based Image Segmentation (Skin Detection)

In this section we describe our methodology for pixel-based skin detection (or image segmentation). As mentioned before, our skin detector runs without arithmetic operations because it is based on Lookup Table SD-LUT. The SD-LUT contains the classification result of any color. The main characteristics of skin detection procedure are:

- To detect skin-color regions in an image, we first transform the source image into HSV image. This involves transforming the R,G,B values at a pixel (x,y) to H,S,V values and saving it at location (x,y) in the HSV image. Figure 4.13(a-b) show an example of RGB image and its corresponding HSV image respectively.
- To reduce the effect of noise, the average filter of a 3 × 3 window is used. To obtain the average color, first the average hue, average saturation, and average value in the 3 × 3 window are determined.
- Usually, input image(s) contain millions of colors. For effective and efficient classification, the number of colors in a source image requires a reasonable reduction. This process is *color quantization*. As mentioned in Section 4.4.3, it is done at Hue channel which is divided into

equal intervals of 6 degrees. The colors in the source image are therefore reduced to 60 primary colors.

- Binary images are used to represent the result of the skin detector. Since we used four-skin models, the output of the skin detector would be four binary images rather than one. The skin regions (objects) in the binary images are represented as a set of connected pixels of value 1 (white), while background is set to value 0 (black). Accordingly, the input image F is transformed into four-binary images represented in different separate layers (or L_1, L_2, L_3, L_4 for abbreviation):

  Layer 1: contains the detected white-skin regions.

  Layer 2: contains the detected shadow-skin regions (or blackish-skin regions).

  Layer 3: contains the detected reddish-skin regions.

  Layer 4: contains the detected light-skin regions.

  The skin detector should pass over all pixels in the source image. For each pixel (x, y) in the source image F, the pixel's color components H, S, and V are used to index the SD-LUT and retrieve the classification result (i.e. its class). Based on the classification result, update the related layer. For example, if the retrieved value from SD-LUT is 2 (means that it belongs to shadow-skin model), the system updates the second layer and set the pixel $L_2(x, y) = 1$, and so on. When the retrieved value is 5 (means that the pixel is non-skin), no action is required because our aim is to detect skin pixels. Each output binary image is called a "skin-map". Therefore, we can say that pixel-based image segmentation generates four-layers of skin-maps as shown in Figure 4.13(c). In this figure, we can notice that the white-skin regions are detected only in layer L_1; while shadow-skin regions are detected in only layer L_2 and so on.

For illustration purpose at this moment, all skin layers are combined using logical OR in one skin-map image named "all in one" as shown in Figure 4.13(d).

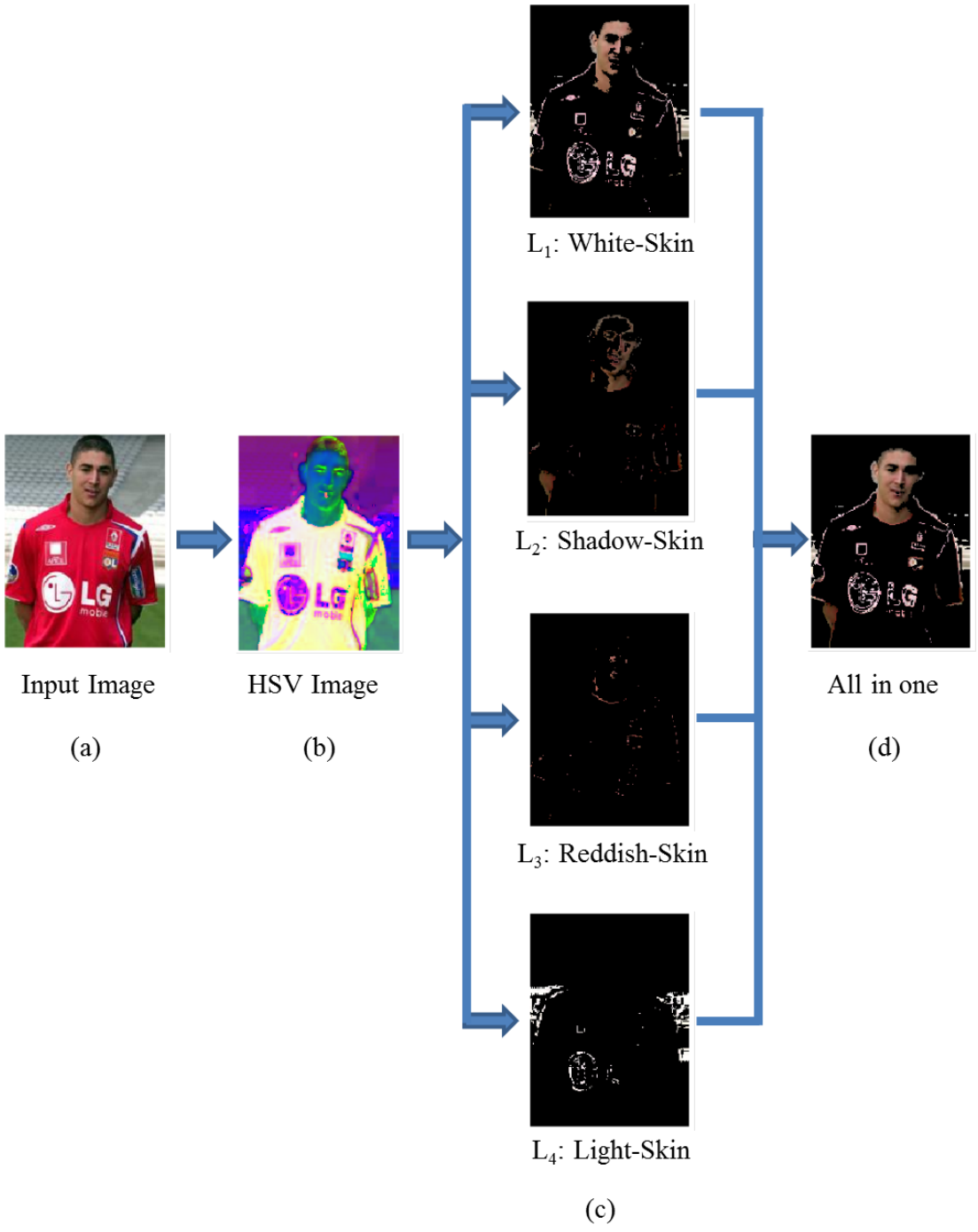


Figure 4.13: Pixel-based image segmentation using multi-skin models; (a) input image RGB; (b) HSV image; (c) Pixel-based image segmentation generates four skin–maps in different layers; (d) all in one.

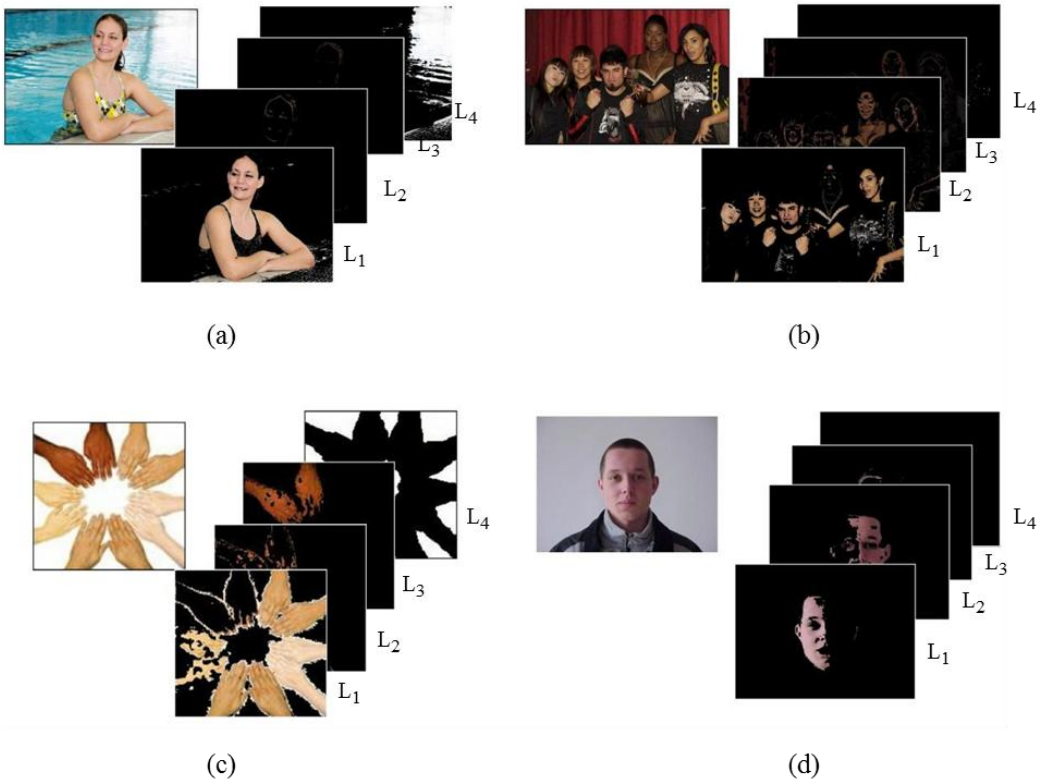


Figure 4.14: Skin detection using multi-skin models approach applied on real images; (a) white-skinned woman, no makeup, and uniform lighting. The white-skin is correctly detected at Layer 1; (b) image with people of different races; the dark-skinned woman is completely detected at Layer 2; (c) various skin tones detected in different layers; and (d) the skin tone of the same individual varies under non-uniform illumination; the left half of the face with normal lighting is correctly classified at Layer 1, while the right half of the face with low lighting is correctly classified at Layer 2.

Figure 4.14 shows more skin detection examples using our approach applied on real images. Figure 4.14(a) shows the output of skin color segmentation for a white-skinned woman with no makeup and under uniform lighting. The white skin is completely detected at Layer 1. Figure 4.14(b) shows the effectiveness of applying multi-skin color models on images containing people of different races. In this figure, the woman with dark skin is completely detected at Layer 2 (although she is misclassified at Layer1) because her skin is evidently belongs to blackish-skin model. Figure 4.14(c) shows an example of various skin tones for different racial groups. The dark reddish hands are correctly classified at Layer 3 only. Although the hands that tend to be yellow are partially classified at Layer 1, the convex-hull algorithm can greatly recover the missing regions in such cases (see Section 6.2.1). Figure 4.14(d) shows that the skin-tone of the same individual varies under non-uniform illumination. The left half of the

face with normal lighting is correctly classified at Layer 1, while the right half of the face with low lighting is correctly classified at Layer 2. Keeping the right half in a separate layer enables us to perform post-processing steps such as correcting the skin color (e.g. lightening).

4.8 Region-Based Segmentation (or Iterative Merge)

This section presents our methodology to enhance pixel-based image segmentation results using iterative merge process. The main drawback of pixel-based image segmentation is that it groups together pixels according to some global attribute. Since pixel-based images represents at best an approximation to the continuous real scene being represented, and since classification functions classify each pixel as either part of the region of interest (i.e. skin) or the background, only a finite level of accuracy can be achieved (Russ, 2007). For example, two pixels at opposite corners of an image may be both detected if both have similar color, even though they are probably not related in any meaningful way. On the other hand, two neighboring pixels may be classified to different classes although they belong to the same region. The reason lies in the fact that each colored pixel is treated individually in relation to the color space (skin or non-skin pixel) without any consideration for the content of neighboring pixels.

This problem can be tackled by using a segmentation algorithm which takes spatial information into consideration. Spatial information is useful because most segments corresponding to real world objects consist of pixels which are spatially connected. This is the basis of region-based segmentation technique. Region-based segmentation is more successful at isolating individual object because they take into account the fact that pixels belonging to a single object are close to one another. For example, consider a human face in an image, the skin of each human face has certain homogeneity between its pixels that could differentiate it from other objects.

Region growing algorithms starts from the bottom, i.e., individual pixel level or small region, and works upward. The starting pixels (or region) are called *seeds* or *seed points*. In many applications, seed pixels can be identified interactively by the user. With the goal of detecting

human faces automatically, seed points should be identified automatically. Therefore, considering the benefits stated above, in this research work pixel-based segmentation is combined with region-based segmentation.

Our approach in this stage is based on simple heuristic such as the fact that human faces in images always retains some regions (or pixels) that reserve the white or light skin tone (e.g. forehead, nose bridge, cheeks). In this research implementation, the detected skin pixels at layer L_1 already correspond to white-skin color. Therefore, these pixels (or regions) are considered as seeds and from these we can start to gather more skin regions from layers L_2, L_3, L_4 in order to create compact candidate face regions.

The neighbouring pixels in the other layers are examined one at a time and added to the growing region if they are sufficiently similar. Again, the comparison can be made with the entire region or just with the local pixels. To ensure that this method allows gradual variations in color intensities, the comparisons are done locally on each pixel rather than the entire region. The parameters are updated during the iterative region growing. A pixel is added to a region if:

- it is adjacent to some pixel of a growing region.
- it belongs to one of the other layers' regions.
- it satisfies the similarity of mean color intensity of the growing region.

It is considered that each seed pixel has 8-connectedness neighbors. But it is not always like this (e.g. it could have less neighboring pixels). The system calls a specific function (procedure) that returns a list of the neighboring pixels based on the coordinates (x, y) and the contents of skin-maps. The list is examined one by one and added to the growing region if they are sufficiently similar. The similarity of two colors is based on the *Euclidean distance.* Initially, we compute the "average" or "mean" color M of all points that belong to the growing region. In the HSV model the *Euclidean* distance between P and M is as follows, adopted by (Gonzalez *et al.*, 2007):

$$
\begin{aligned}
D(P,M) &= \|P-M\| \\
&= \left[(P-M)^T (P-M)\right]^{1/2} \\
&= \left[(P_H - M_H)^2 + (P_S - M_S)^2 + (P_V - M_V)^2\right]^{1/2}
\end{aligned} \tag{4.1}
$$

where the $\|.\|$ is the norm of the arguments, and subscript H, S, and V denote the color components in HSV color space. Then, we can say that a point P is similar to M if the *Euclidean distance* $D(P,M)$ between them is less than or equal to a certain threshold *T*. The threshold of color intensity used is T = 0.5 (identified experimentally). The unmerged background regions in layers L_2, L_3, L_4 are rejected at the early stage of computation.

Some skin detection examples are shown in Figure 4.15. More skin detection results, evaluations, and comparisons are illustrated in Sections 4.11 and 4.12 respectively.

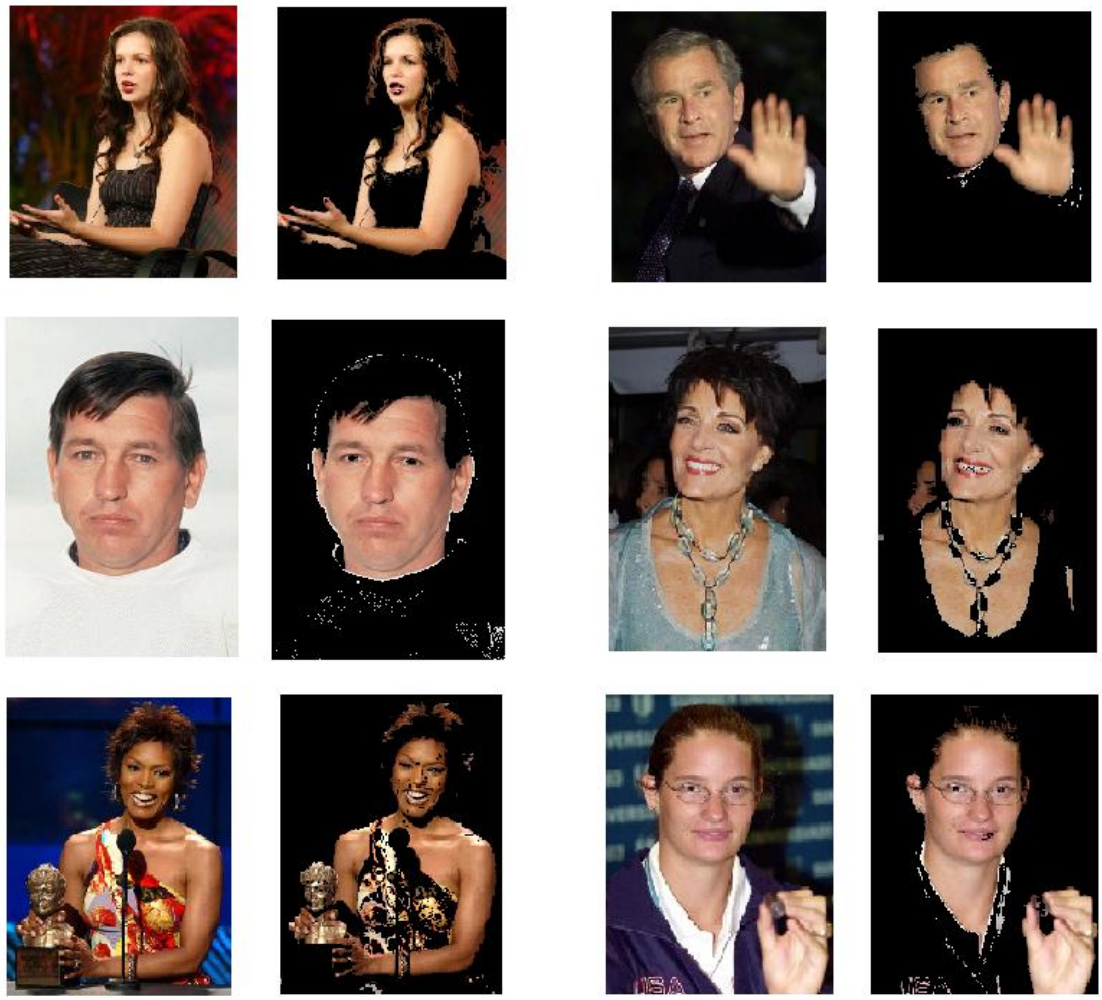


Figure 4.15: Skin detection examples using the proposed method.

4.9 Skin-Color Modeling and Classification Boundaries

In this research we sought to use an algorithm that would take the measured features of an unknown pixel (i.e., three-color components H, S, and V) as input and then predict the true class membership as output (i.e. White-skin, Blackish-skin, Reddish-skin, Light-skin, or Non-skin). The rate of false predictions depends on numerous factors, such as the degree of overlapping among the classes in the feature space, the amount of noise data, and the generality of training samples.

The goal is to build a skin color model that defines classification boundaries (or decision functions) to discriminate between different classes. The success or failure of image segmentation depends mainly on the choice of appropriate classification boundaries. In an ideal case, classification boundaries could completely separate different classes. Simpler classification boundaries are favored over those that are needlessly complicated. Generally, building complex models with complex classification boundaries often leads to lower accuracy classifiers (Duda et al., 2001).

An obvious and familiar solution is to adjust the decision boundaries manually until acceptable results are achieved. However, this is not possible in cases where fully automatic segmentation is required. Alternatively, it might be possible to determine in advance the fixed decision boundaries that will always give best results. In practice, working with complex images makes the problem harder. In real-world implementation, different methods usually produce different classification boundaries. It is very difficult to know which classification method (or which skin model) will be the best for all kind of images. To perform a fair practical evaluation of image segmentation methods (or skin color models), a standard testing and evaluation procedure should be devised. Next section is dedicated for this purpose.

4.10 Testing and Evaluation of Image Segmentation Methodologies

An important characteristic underlying the design of image segmentation methodologies is the considerable level of testing and evaluation that is required before arriving at the final acceptable solution (Gonzalez *et al.*, 2007) .

Testing and evaluation step gives us tools to measure and compare the characteristics of segmentation methodologies, and thus determines their performance. Numerous methods for image segmentation were shown in the literature (see Chapter 3). It is important to be able to evaluate and compare these methods. Evaluation is important not only for application developers, who need to select the correct tool for the job, it is also important for researchers to accurately evaluate their methodology in order to improve the results. It also allows researchers to justify new methods via formal comparison with existing methods. This implies the need to formulate testing approaches that, in general, can reduce the cost and time required.

Up to date, although there is an enormous amount of research dedicated to image segmentation algorithms, there is a limitation about how to measure segmentation accuracy and error rates (Gonzalez *et al.*, 2007; Russ, 2007). The traditional method is done based on simple criteria of the percentage of pixels misclassified using ground truth images (see Section 4.10.3.2). Another common method for evaluating the effectiveness of a segmentation method is subjective evaluation, in which a human visually compares the image segmentation results for different segmentation algorithms, which is a tedious process and inherently limits the depth of evaluation to a relatively small number of segmentation comparisons over a predetermined set of images (Zhang , Fritts, & Goldman, 2008).

In most previous works, researchers perform tests and experiments using their own test images. However, *"Evaluation of segmentation algorithms thus far has been largely subjective, leaving a system designer to judge the effectiveness of a technique based only on intuition and results in the form of a few example segmented images"*(Unnikrishnan, Pantofaru, & Hebert, 2007).

Some methods are designed for a specific purpose in their approach and are therefore inappropriate to be adapted for other applications without considering some underlying assumptions. Others have one or more tunable parameters which must be adjusted by the user, rather than learned automatically from the image itself.

Therefore, it is not easy to compare different image segmentation algorithms. As a consequence it is still very difficult to answer the question "How good is a given algorithm?". The factors that affect the evaluation of a method can be summarized as follows:

- Each method uses different test images. Therefore, there is lack of standard test images.
- The way of processing and final goals in these methods varies because they are intended to be used in different environments. The question is: "How can general objective criteria be formulated for comparison?"
- Each method uses different raw data sets (e.g. skin samples or training data) which are collected manually by the researchers. The size and quality of raw data have a direct effect on the skin color cluster even when using the same method. If new raw data of skin samples are collected from another database, the skin color cluster may change dramatically.
- Using different methods for modeling skin color clusters yields different classification boundaries (even with the same raw data). It is important to evaluate the feasibility of classification boundaries prior to any further testing steps.
- It is very difficult to know which algorithm will be the fastest for a given problem. It depends on many factors, including the complexity of the classification boundaries, the design of the classifier, the acceptable error goal, etc.

Therefore, getting fair evaluation of these methods becomes difficult and most surveys refer to the results obtained by the authors.

In general, to perform a fair practical evaluation of the skin segmentation methods, a standard set of test images must be used. These test images should contain skin samples that are regarded as a generalized representative data set.

Initially, we tried to collect a set of standard test images for evaluation purpose. Since complex images come from different sources (i.e. databases, internet, etc.) and types (i.e. indoor or outdoor, single face or multiple faces, same size or different size, etc.), it is difficult to decide which image is a good representative image. So, instead of collecting real images from different sources manually, standard test images can be created automatically (i.e. generated). It is based on general fact: If the test images that cover all skin tones can be found (or created), then the evaluation process will be fair and standard.

In this thesis, a novel method is proposed for testing and evaluating the skin segmentation methods. Our method consists of three steps:

- Propose a set of standard test images. These images are generated automatically from color space.
- Establish general guidelines for evaluating the feasibility of the classification boundaries (i.e. the skin color model) which are used for image segmentation.
- Propose step-by-step procedure for testing and evaluation.

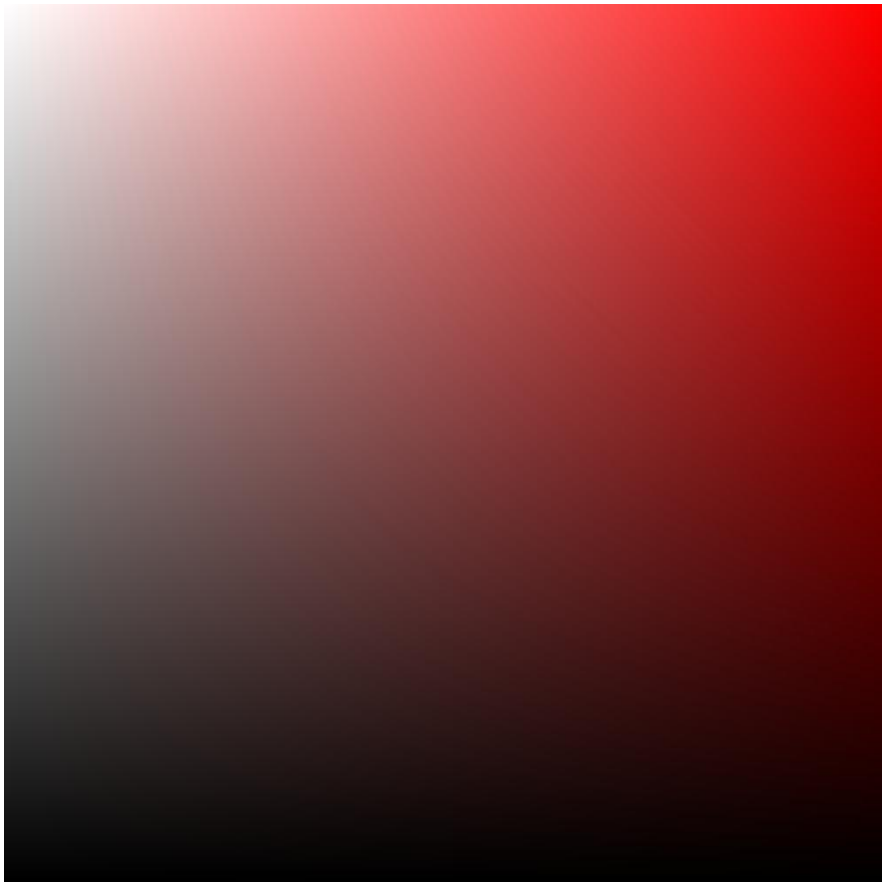
4.10.1 Proposing Standard Set of Test Images

The key contribution of this method is based on creating a standard set of test images for evaluation purposes. Each test image contains almost 100% of all the tones of a specific skin color with smooth gradual change at each axis. These test images contain no faces, no hands, and no human targets at all, but they contain most of skin color tones as shown in Figure 4.16 and Figure 4.17. Furthermore, these images are free of noise. The main idea of proposing these images is as follows: these images can detect the classification boundaries of any skin model independent of the color space or the methodology. Plotting classification boundaries is the best

way to understand them. Then, the detected classification boundaries would be evaluated (Section 4.10.3.1). As shown in Figure 4.16, our system generates 60 standard test images for this purpose. Although these images are saved in RGB format, they are generated from the HSV color space using quantized Hue channel (i.e. slices). The hue H component is constant (i.e. for each test image) while saturation S and value V are variables representing the x-axis and y-axis respectively. Although, some of these images can be found in some literatures for illustration purposes, we are the first in this field to use these images as a tool to measure and compare the characteristics of segmentation methodologies in order to determine their performance. Figure 4.17 shows four standard test images in bigger size for illustration purpose; these are at Hue=00°, 06°, 12°, and 18°. In this figure, one can notice clearly that each image contains all the tones of a specific color (hue) with smooth gradual change. These test images cannot be generated directly from RGB color space, because each of the R, G, and B components contains both color and intensity information, whereas, generating these images from HSV color space can be done directly; then transformed to RGB images. The efficiency and advantages of these standard test images are shown in Section 4.10.3.1.

Figure 4.16: The proposed standard set of test images used as a tool for testing and evaluating different segmentation methods in application to skin detection.

Hue=00°



Hue=06°

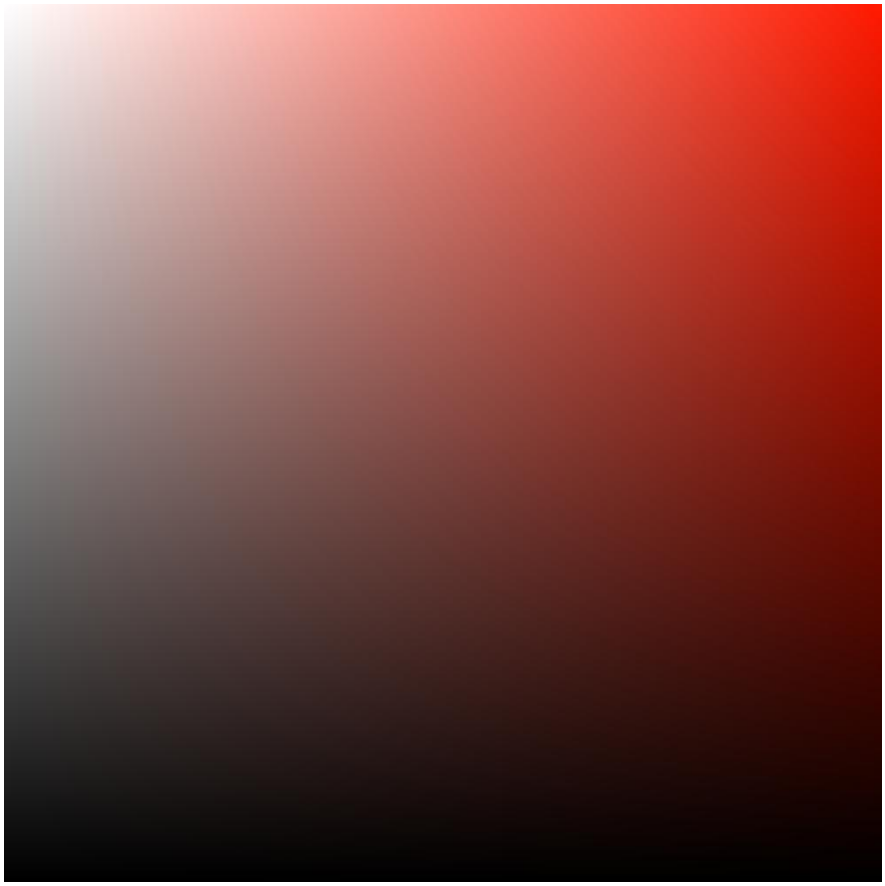
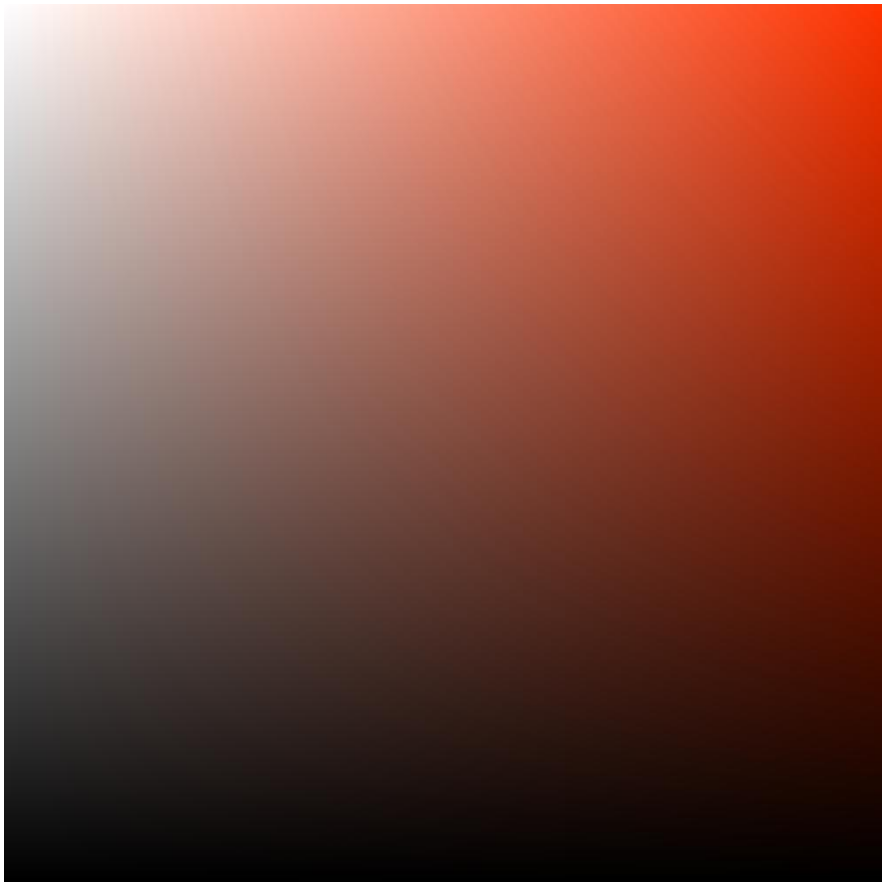


Figure 4.17: Example of standard set of test images.

Hue=12°



Hue=18°

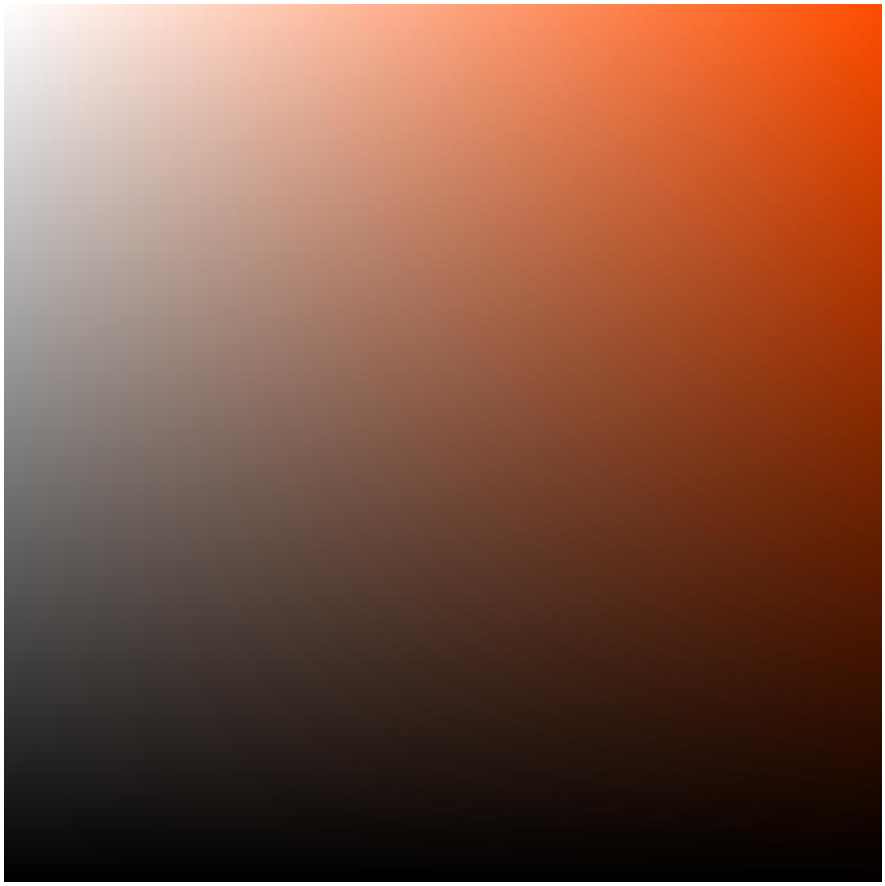


Figure 4.17 (*continued*): Example of standard set of test images.

4.10.2 Guidelines for Evaluating the Feasibility of Classification Boundaries

The goodness of segmentation methodology is all about evaluating the feasibility of classification boundaries. This evaluation aims to justify the practicality of classification boundaries for a given method. Since different skin-color models lead to different classification boundaries, the classification boundaries should first be evaluated before being applied on real images. From our point of view in this research, using real images cannot be useful for testing and evaluating the feasibility of classification boundaries. Being able to accurately gauge the performance of classification boundaries gives an insight into what constitutes good boundaries. The best way to justify these boundaries is by visual representation.

To illustrate this point, an interesting example can be used as shown in Figure 4.18, adopted from (Duda *et al.*, 2001), which shows a complex classification boundary for two-class classification problem. Although such complex classification boundary may lead to perfect classification of the current training samples, it also results in poor performance on the novel pixels, that is, the pixels not yet seen. For example, in this figure; the novel test point marked ? evidently comes from class 1, but the complex decision boundary leads it to be misclassified as class 2. With such solutions though, our satisfaction would be premature because the central aim of scheming the classification boundaries is to suggest actions when presented with novel patterns, i.e., pixels not yet seen (Duda *et al.*, 2001).

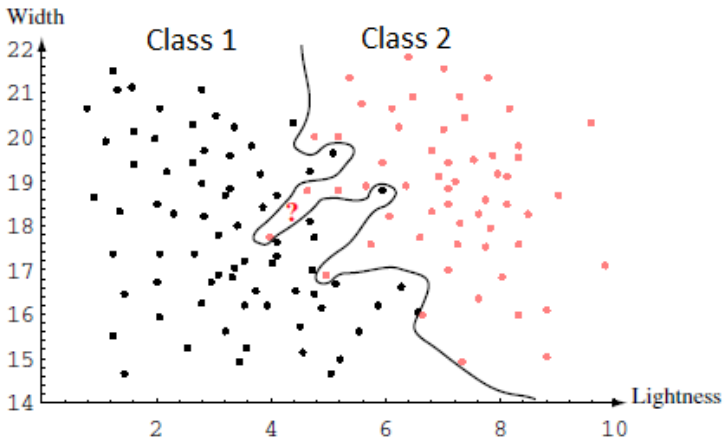


Figure 4.18: Complex model for two-class problem, leading to classification boundaries that are complicated.

These classification boundaries are not likely to provide a good generalization as they seem to be "tuned" to the particular training samples, rather than some underlying characteristics or true models of the classes that have to be separated. For example, given a classification boundary with thin gulfs (as show in Figure 4.18) is not practical for the task of color-based image segmentation. It is inconceivable that while some regions in a small neighborhood belong to the skin, the other regions that fall in between do not. This is due to the fact that adjacent entries in a color space show very similar colors. Therefore, there may be a need to simplify the classification boundaries (e.g. removing gulfs), motivated by a belief that the underlying skin models will not require a decision boundary that is complex[1] (Duda *et al.*, 2001).

Haralak and Shapiro (1984), as stated by (Pratt, 2001), established the following qualitative guideline for good image segmentation:

> "*Region interiors should be simple and without many small holes. Adjacent regions of segmentation should have significantly different values with respect to the characteristic on which they are uniform. Boundaries of each segment should be simple; not ragged, and must be spatially accurate".*

The above guidelines are useful for general purpose image segmentation as well as independent on specific test images. What is sought is a formulating guideline for a specific purpose that is detecting human skin using color feature and based on specific set of standard test images.

In this work, five guidelines are proposed to evaluate the feasibility of classification boundaries, as follows:

(a) The classification boundaries should yield compact regions. By compact region is meant no broken region (i.e. each class should yield no more than one region), no bridge gaps, and without holes.

---

[1] The philosophical underpinnings of this approach derived from William O. Occam (1284-1347c), who advocated favoring simpler explanations over those that are needlessly complicated "Entities are not to be multiplied without necessity"(Duda et al., 2001).

(b) The regions should match with the real distribution of training data.

(c) The region's contour should be smooth, simple, without thin gulfs, and without protrusions.

(d) The region's shapes should not be overlapped (e.g. snaky, complex, zigzag, etc.).

(e) The shape of regions for adjacent slides should be changed gradually in the 3D space that forms a sold 3D body. Abrupt change in the shape through successive adjacent slides is undesirable.

4.10.3 Step-by-Step Procedure for Testing and Evaluating Skin Segmentation Methods

In this research, we propose the following steps: evaluating the feasibility of classification boundaries using standard set of test images, quantitative evaluation, and qualitative evaluation.

4.10.3.1 Evaluating the Feasibility of Classification Boundaries

This step aims at evaluating the feasibility of classification boundaries. To the best of our knowledge, pervious works use real images for this purpose. In this research, the evaluation is based on using the proposed standard set of test images with the general guidelines shown in Section 4.10.2. To illustrate this point, let us consider skin segmentation results using a real image as shown in Figure 4.19(a), adopted from (Kovac *et al.*, 2003). The source image is shown in the left column. The skin segmentation output is shown in the right column. As shown in this figure, the segmentation output does not tell us any information about the feasibility of classification boundaries. Using more real images will not resolve this issue. On the other hand, Figure 4.19(b) shows the segmentation output of applying Solina's method (2003) on one of the proposed standard test image. In this figure, the classification boundaries had been detected and plotted graphically. Figure 4.19(c) shows the results of applying Garcia's method (1999) using the same standard test image. In these two figures (b-c), it can be seen that although the same test image is used, we got two different classification boundaries. This difference can provide us with information about the goodness or weakness of each method. Using more standard test

images can gives us more and more information. More details and findings are described in the next sections.

As these images are generated with gradual smooth change in color-tone, the classification boundaries can be plotted accurately. When the classification boundaries are found infeasible in some situations, the method should be revised. For example, if classification boundaries yield non-compact regions that contain gulfs or produce regions that do not match the real distribution of the training data, then the classification boundaries (or skin color model) should be adjusted.

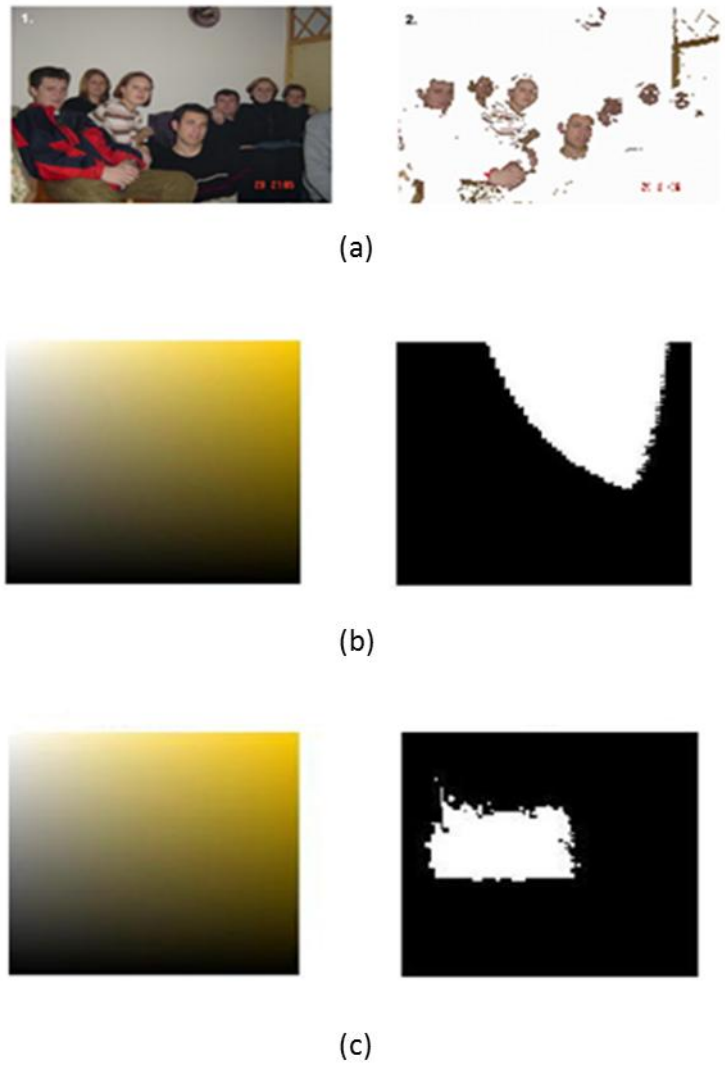


Figure 4.19: Skin segmentation using different test images; (a) applying Solina's method on non-standard test image; (b) Applying Solina's method on a standard test image; (c) Applying Garcia's (1999) method on the same standard test image.

4.10.3.2 Quantitative Evaluation

This step aims at evaluating the performance of classification boundaries using the raw data and/or ground truth images in terms of statistical measures, namely: TP, TN, FP, FN, FPR, FNR, Accuracy, and Recall (or sensitivity) as follows (Taqa & Jalab, 2010; Matlab 2010):

$$\begin{aligned} &\text{False negative rate} \quad FNR = FN/(TP+FN); \\ &\text{False positive rate} \quad FPR = FP/(TN+FP); \\ &\text{Accuracy} = (TP+TN)/(TP+FN+FP+TN); \\ &\text{Recall or Sensitivity} = TP/(TP+FN) \end{aligned} \tag{4.2}$$

where TP is the number of true positive instances; TN is the number of true negative instances; FP is the number of false positive instances; FN is the number of false negative instances.

As mentioned before, the raw data consists of more than 20,000,000 pixels. The quantitative evaluation based on raw data is shown in the next sections. Quantitative evaluation using ground truth images refer to a process in which a pixel on a test image is compared to what is there in reality in order to determine the accuracy of the image segmentation approach. In this research, the Adobe Photoshop CS3 software is used to prepare the ground truth images. Figure 4.20 shows examples of ground truth images. As shown in this figure, the faces, hands, shoulders, and other exposed parts of the human body are identified and isolated from other objects; and then, all other objects in the image are set to black color (i.e. background). The new generated image is used as ground truth to evaluate the results of the segmentation methods.

It is common practice to create a confusion matrix in which the classification results are compared to the ground training data. The strength of a confusion matrix is that it identifies the nature of the classification errors, as well as their quantities. In a confusion matrix, the (i,j) element in the confusion matrix is the number of samples whose known class label is class ω_i and whose predicted class is ω_j where $i \neq j$ (i.e. misclassified). The diagonal elements (i,i) represent correctly classified data.

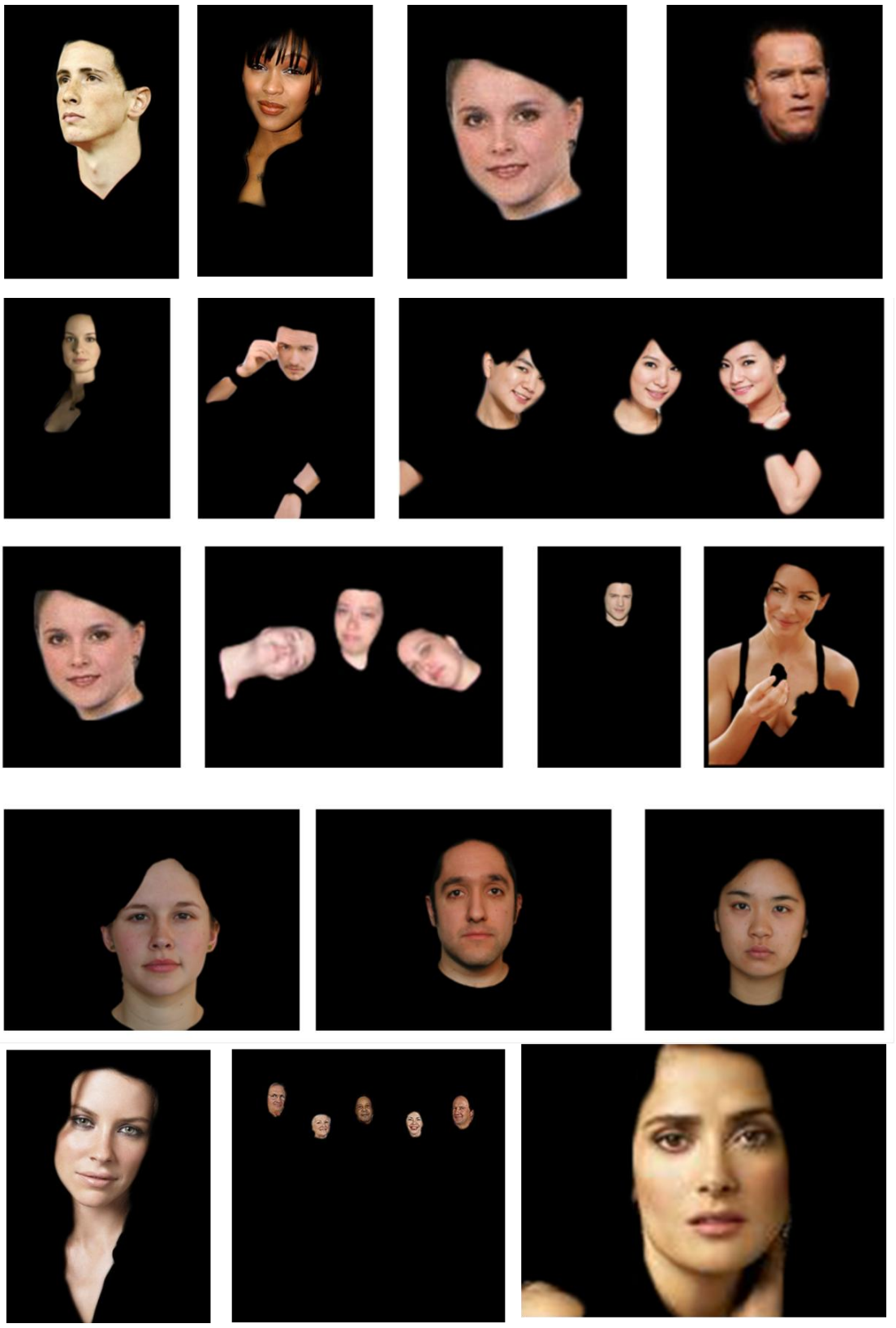


Figure 4.20: Examples of ground truth images.

4.10.3.3 Qualitative Evaluation

This step is subjective evaluation in which a human visually evaluate the image segmentation results using real images.

4.10.4 Applying the Proposed Testing and Evaluation Procedure to Other Works

This section presents applying the proposed testing and evaluation procedure to seven skin color modeling and detection methods. These methods are:

- Solina, *et al.* (2002) method - Explicit thresholds using RGB color space.
- Chen and Wang (2007) method - Explicit thresholds using RGB model.
- Baskan *et al.* (2002) method - Explicit thresholds using HSV model.
- Garcia & Tziritas (1999) method - Explicit thresholds using HSV color space.
- Bayes Classifier based on two-class classification problem.
- Bayes Classifier based on Multi-skin Models.
- Linear Discriminant Analysis (LDA).

4.10.4.1 Solina, et al. (2002) Method - Explicit Thresholds using RGB color space

Solina, et al. (2002) defined explicit thresholds to describe skin color cluster in RGB color space using the following rules:

$$\begin{aligned} &R > 95 \text{ and } G > 40 \text{ and } B > 20 \text{ and;} \\ &Max(R, G, B) - min(R, G, B) > 15 \text{ and;} \\ &|R - G| > 15 \text{ and } R > G \text{ and } R > B \end{aligned} \tag{4.3}$$

To evaluate the feasibility of these rules we would apply them on the proposed standard set of test images. The results of image segmentation performed by these rules are shown in Figure 4.21. Figure 4.21(a) shows the original image; Figure 4.21(b) shows skin segmentation output (i.e. the detected classification boundaries). Although, the classification boundaries are acceptable in many cases, but there are some cases lead to incorrect classification results.

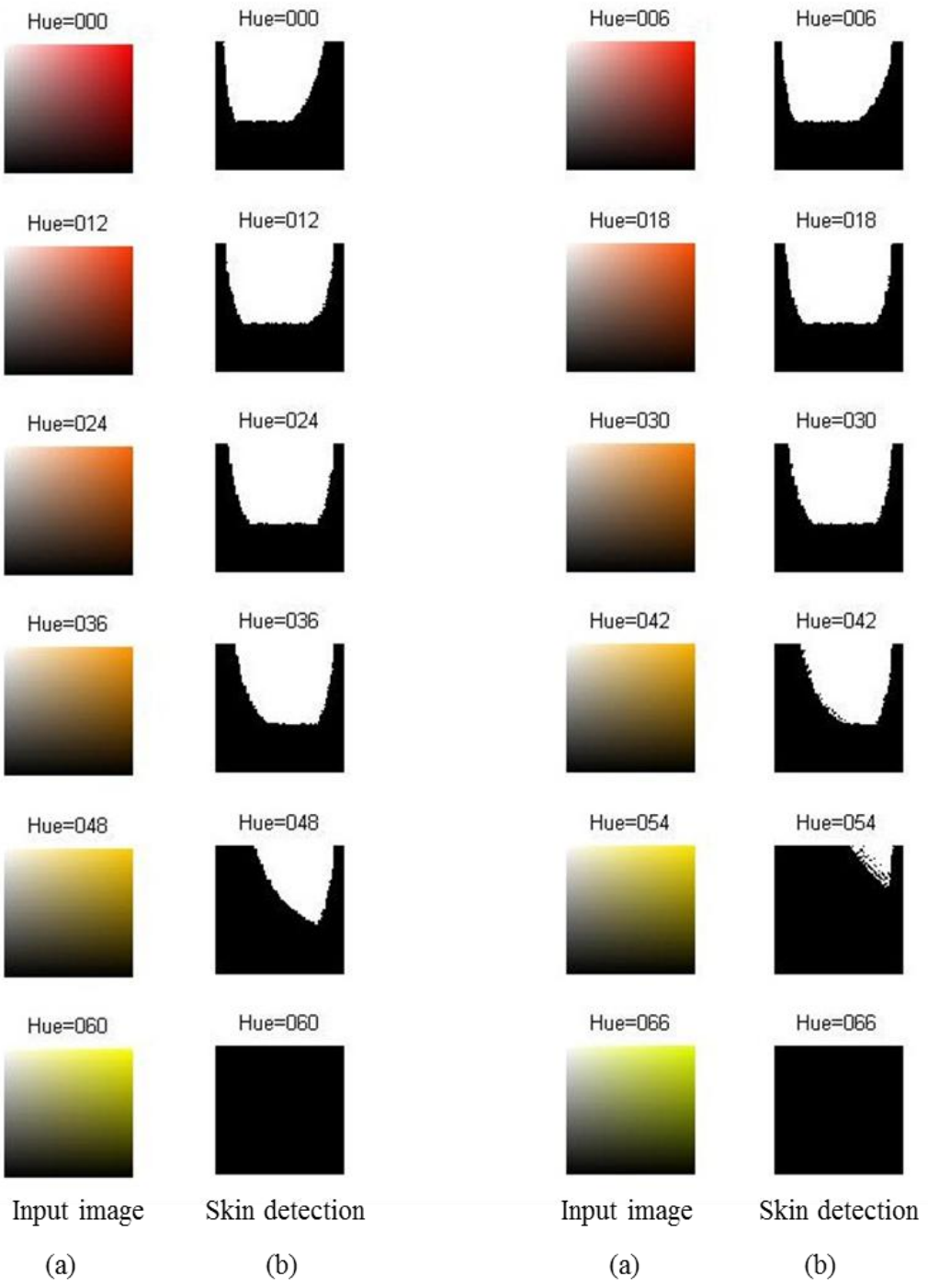


Figure 4.21: Skin detection results using Solina's method (2003) applied on standard test images; (a) input image; (b) skin detection output.

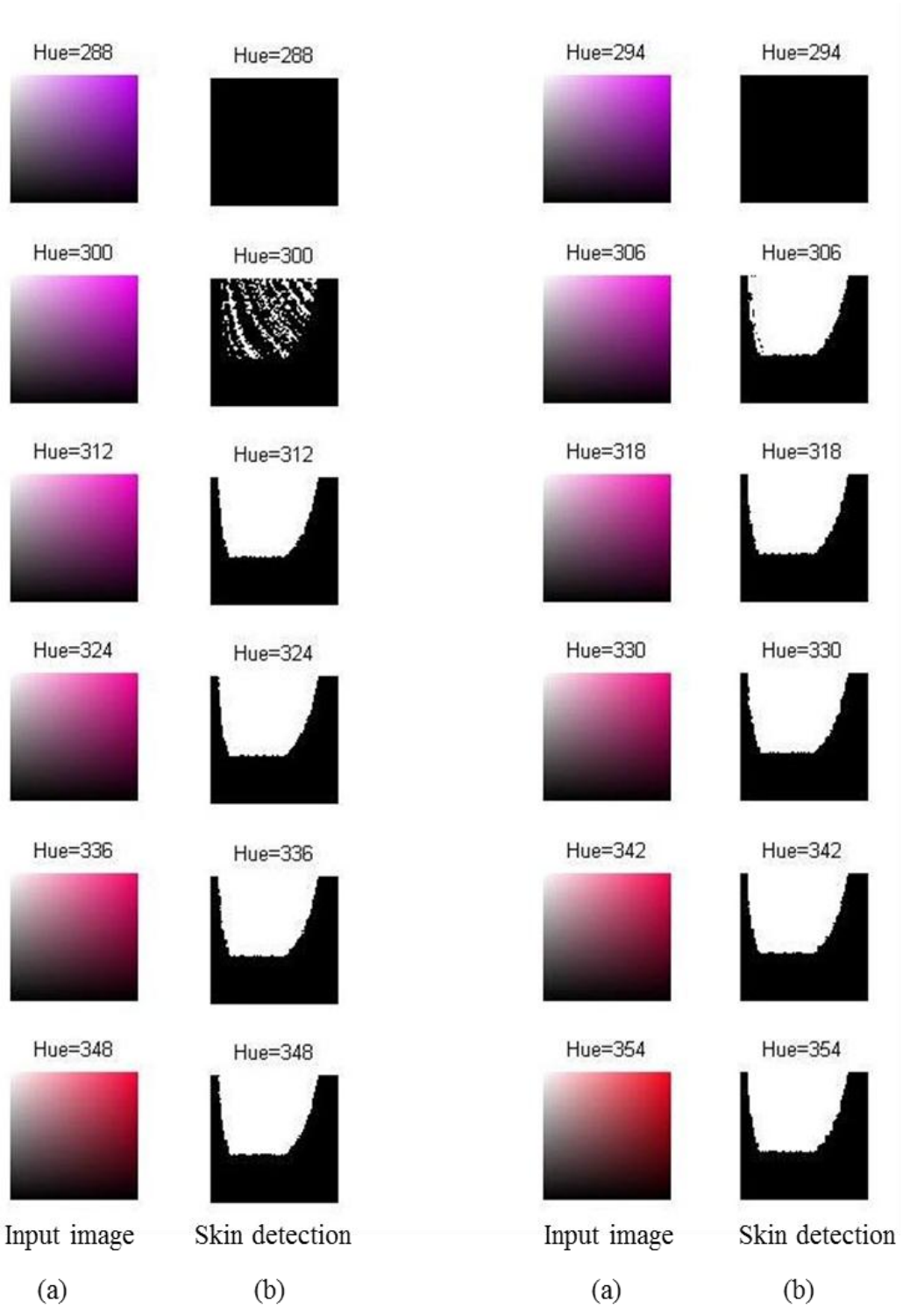


Figure 4.21 (*continued*): Skin detection results using Solina's method (2002) applied on standard test images; (a) input image; (b) skin detection output.

Two examples of FP errors are explained in Figure 4.22 using two test images at hue=0° and 312°. In this figure, the pixel marked ? evidently belongs to non-skin class, but the classification boundaries incorrectly classified it as skin. In the first row, the pixel tends to high concentrated red while the second row shows that the pixel tends to high concentrated violet.

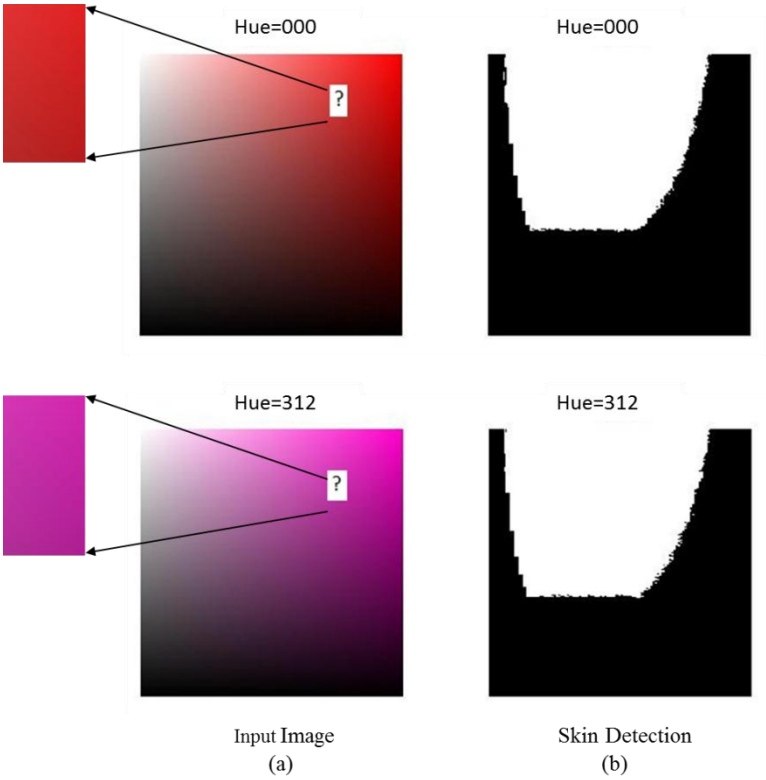


Figure 4.22: Skin detection results using Solina's method (2002) applied on standard test images; H=0 and 312; (a) input image; (b) skin detection output. In each row, the pixel marked ? is classified as skin while it evidently belongs to non-skin.

Figure 4.23 illustrates both FN & FP errors in the same test image. In this test image, it is clear that the left hand side of the image is likely to be skin color more than the right hand side. The first row shows an example of FN; the pixel marked ? evidently belongs to skin class, but the classification boundaries incorrectly classified it as non-skin. The second row shows an example of FP; in this figure, the pixel marked ? evidently belongs to non-skin class, but the classification boundaries incorrectly classified it as skin.

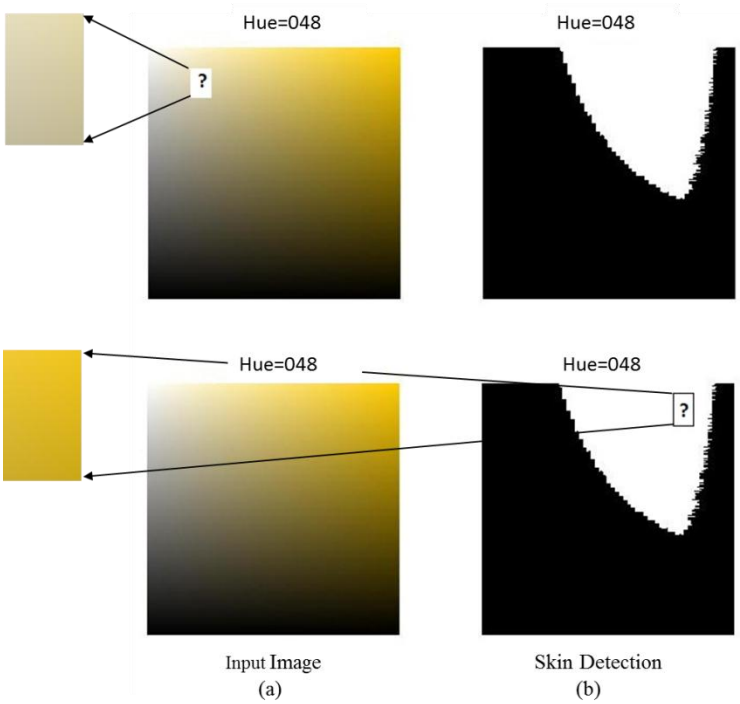


Figure 4.23: Skin detection results using Solina's method (2002) applied on standard test images; H=48. Left column shows the original image; right column shows skin segmentation output. The first row shows an example of FN; the pixel marked ? evidently belongs to skin class, but the classification boundaries incorrectly classified it as non-skin; the second row shows an example of FP; the pixel marked ? evidently belongs to non-skin class, but the classification boundaries incorrectly classified it as skin.

The qualitative evaluation of Solina's method using real images is shown in Figure 4.24. This figure shows both kinds of FP & FN examples. Figure 4.24(a) shows two examples of FP errors. In this figure, it is clear that the shirts in both images are non-skin but they are incorrectly classified as skin pixels. Figure 4.24(b) shows two examples of FN errors in which skin pixels are missed in both images.

The quantitative evaluation of Solina's method using our training data is shown in Table 4.1. As shown in this table, the accuracy rate of this method is 85.703% with high FNR and FPR of 6.585% and 19.492% respectively.

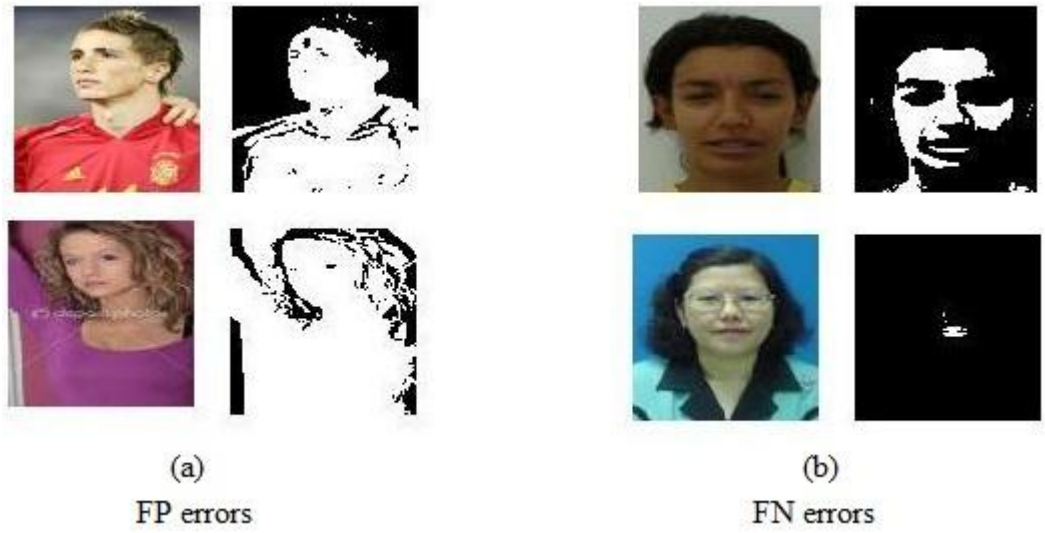


Figure 4.24: Skin detection results using Solina's method (2002) applied on real images; (a) examples of FP errors. Although the shirts are tend to be non-skin, they are wrongly classified as skin pixels; (b) examples of FN errors in which skin pixels are wrongly classified as non-skin pixels.

Table 4.1: Pixel-based quantitative results of Solina's method using our training data.

No. of Pixels	TP	TN	FP	FN	FNR %	FPR %	ACCURACY %	RECALL %
24,328,670	9,147,005	11,703,299	2,833,539	644,827	6.585	19.492	85.703	93.415

4.10.4.2 Chen and Wang (2007) Method - Explicit Thresholds using RGB Model

In the work of Chen and Wang (2007), the set of decision rules was constructed empirically. Assume that each pixel color is represented by a 3-D vector (*R*, *G*, *B*). Pixels that satisfy the following rules are classified as skin pixels:

$$\begin{aligned} &R > G \text{ and } G > B \text{ and;} \\ &R > 95 \text{ and } G > 40 \text{ and } B > 20 \text{ and;} \\ &30 < (R - G) < 80 \text{ and } (R - B) < 120 \text{ and;} \\ &10 < (G - B) < 80 \text{ and } (G + B - R) > 10 \end{aligned} \tag{4.4}$$

To evaluate this method we would apply these rules on the proposed set of test images. The results of image segmentation performed by this method are shown in Figure 4.25.

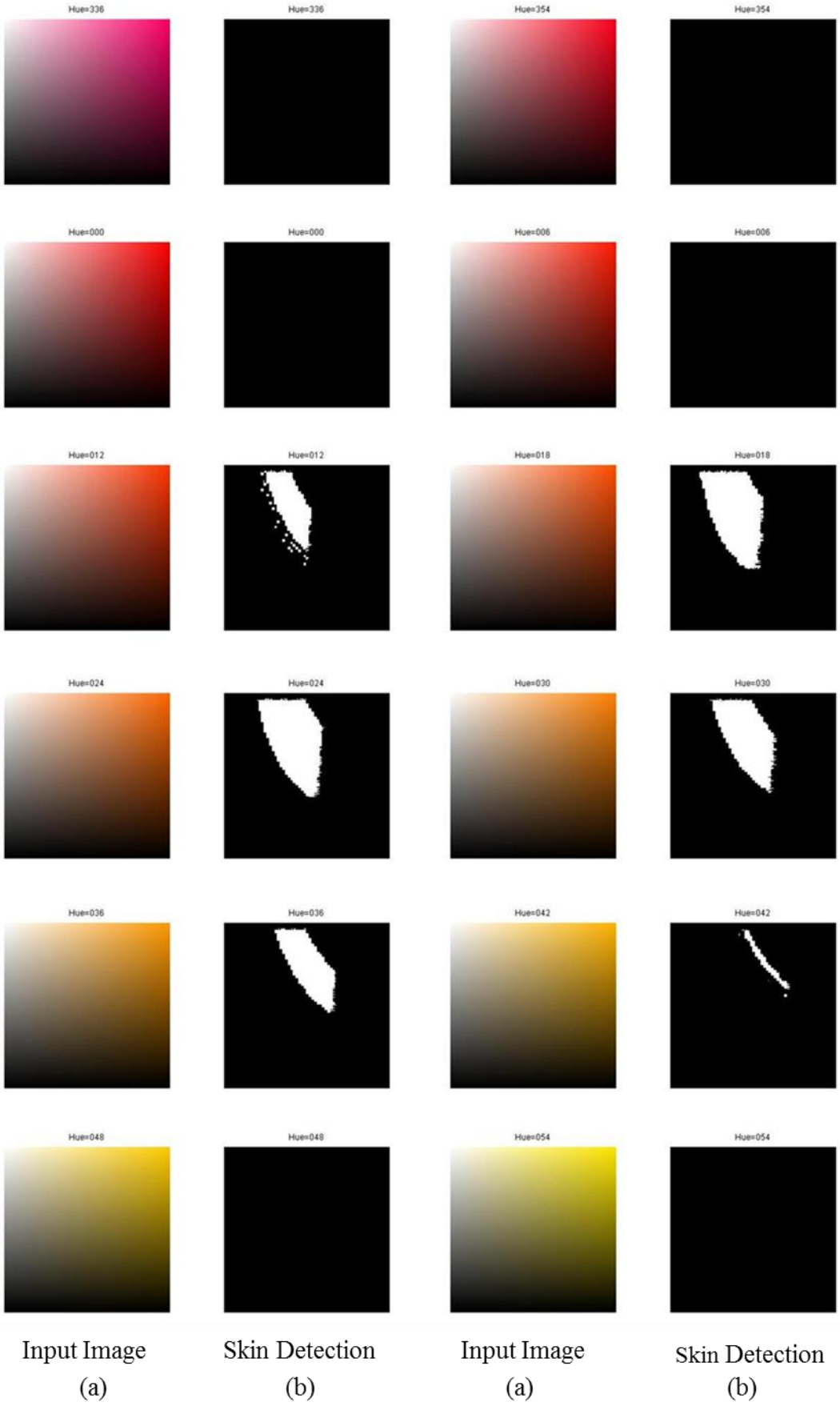


Figure 4.25: Skin detection results using Chen and Wang (2007) approach applied on standard test images; (a) Input image; (b) skin detection output.

The feasibility evaluation of the classification boundaries shows very low skin detection rate (i.e. FN is high). An example of FN errors is explained in Figure 4.26. In this figure, the pixel marked ? evidently belongs to skin class, but the classification boundaries incorrectly classified it as non-skin. The first row shows no skin detected at all, while the second and third rows show that the detection rate is very low compared to Solina's method (2002).

The qualitative evaluation of Chen's Method (2007) using real images is shown in Figure 4.27. As shown in this figure, this method shows high FN errors in which a large number of human skin pixels are missed (i.e. wrongly classified as non-skin pixels).

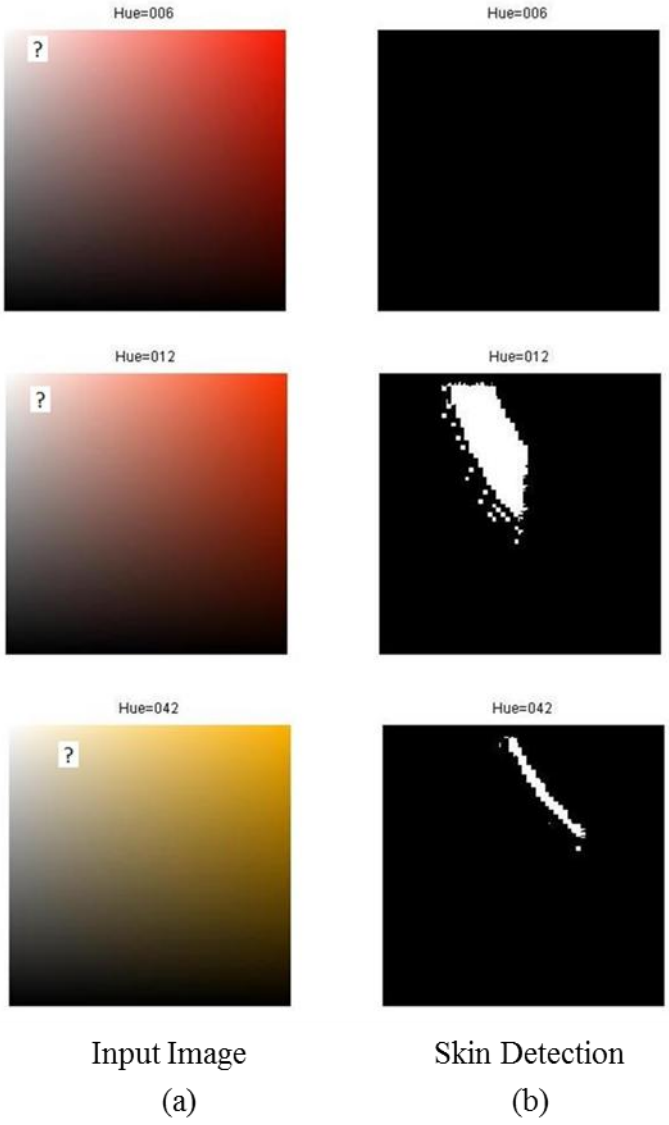


Figure 4.26: Skin detection results using Chen and Wang (2007) approach applied on standard test images, H=6, 12, and 42. (a) input image; (b) skin segmentation output. The pixel marked ? evidently belongs to skin, but the method wrongly classified it as non-skin.

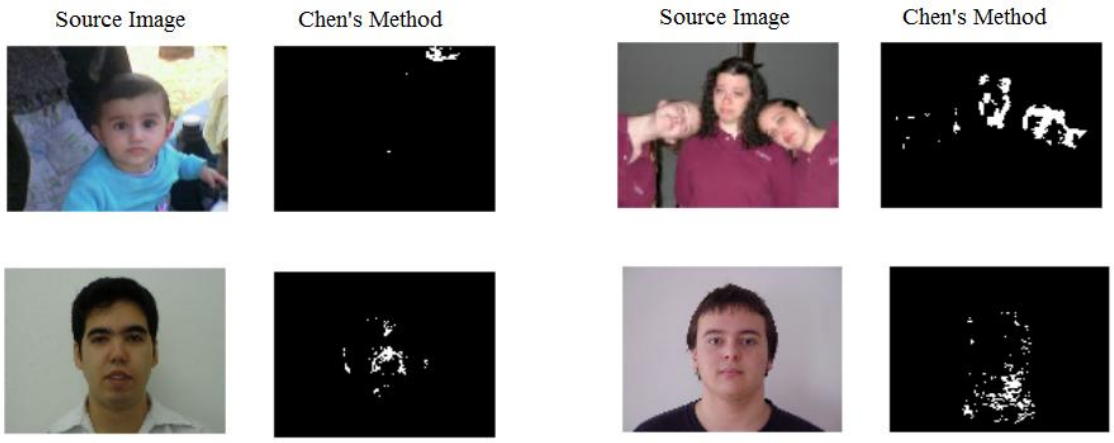


Figure 4.27: Skin detection results using Chen's Method (2007) applied on real images. This method shows high FN errors in which a large number of human skin pixels are wrongly classified as non-skin pixels.

The pixel-based quantitative evaluation of Chen's Method using our training data is shown in Table 4.2. As shown in this table, the accuracy rate of this method is 85.567% with recall of 64.140% and high FNR of 35.860%. As mentioned in the preceding sections FN errors is more critical problem (see Section 4.4.1)

Table 4.2: Pixel-based quantitative evaluation of Chen's method using our training data.

No. of Pixels	TP	TN	FP	FN	FNR %	FPR %	ACCURACY %	RECALL %
24,328,670	6,280,440	14,536,838	-	3,511,392	35.860	0.000	85.567	64.140

4.10.4.3 Baskan et al. (2002) Method - Explicit Thresholds using HSV Model

Baskan *et al.* (2002) employed two skin filters in HSV color space as follows:

$$0.23 \leq S \leq 0.69, \quad and \quad 0° \leq H \leq 40°;$$

$$0.23 \leq S \leq 0.69, \quad 0° \leq H \leq 40°, \quad and \quad S' \geq 0.25 \tag{4.5}$$

To evaluate these filters, the proposed standard set of test images is used. An example of image segmentation results are shown in Figure 4.28 in which the first column shows the original image; second column shows skin segmentation output of Filter-1; the third column shows the

segmentation output of Filter-2. This method outcome the same classification boundaries for all test images (which is unreasonable because these images are completely different in color). Furthermore, the boundaries are parallel to the y-axis (i.e. V component is ignored). If the actual distribution of color values has some other shape in the color space, for instance if it is stretched out in a direction that is not parallel to one axis, then these boundaries are inadequate to select the desired range of the real color distribution.

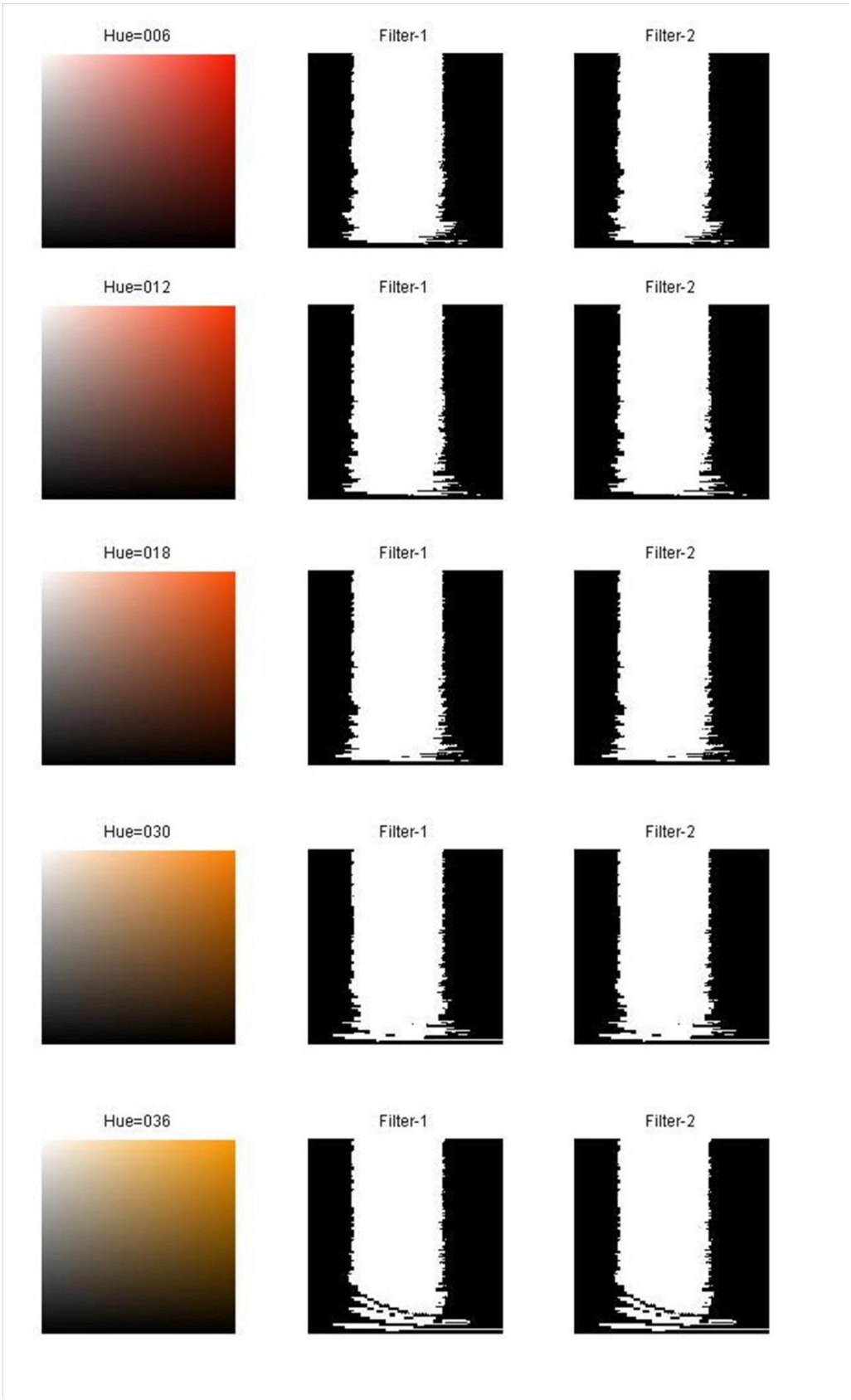


Figure 4.28: Skin detection results using Baskan's Approach (2002). Left column original image; second column shows skin segmentation output of Filter-1. The third column shows skin segmentation output of Filter-2. The approach shows the same region's boundaries for all test images. This means that it forms a cube in 3D color space.

The feasibility evaluation of some classification boundaries is shown in Figure 4.29. In this figure, the first row shows an example of FN error; the pixel marked ? evidently belongs to skin class, but the classification boundaries incorrectly classified it as non-skin. The second row shows an example of FP error; the pixel marked ? evidently belongs to non-skin class, but the classification boundaries incorrectly classified it as skin.

An example of the qualitative evaluation of Baskan's Method using real images is shown in Figure 4.30. This figure shows both kind of FP & FN errors using Baskan's method applied on real images. Figure 4.30(a) shows two examples of FP errors. Figure 4.30(b) shows two examples of FN errors in which skin pixels are missed.

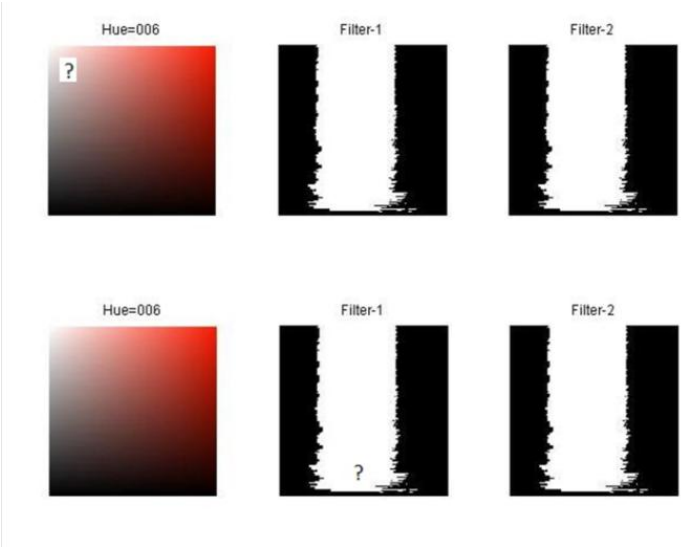


Figure 4.29: Skin detection results using Baskan's Approach (2002) applied on test images H=6. Left column shows original image; second column shows skin segmentation output of Filter-1. The third column shows skin segmentation output of Filter-2. The upper row shows an example of FN; the pixel marked ? evidently belongs to skin class, but the classification boundaries incorrectly classified it as non-skin. The lower row shows an example of FP; the pixel marked ? evidently belongs to non-skin class, but the classification boundaries incorrectly classified it as skin.

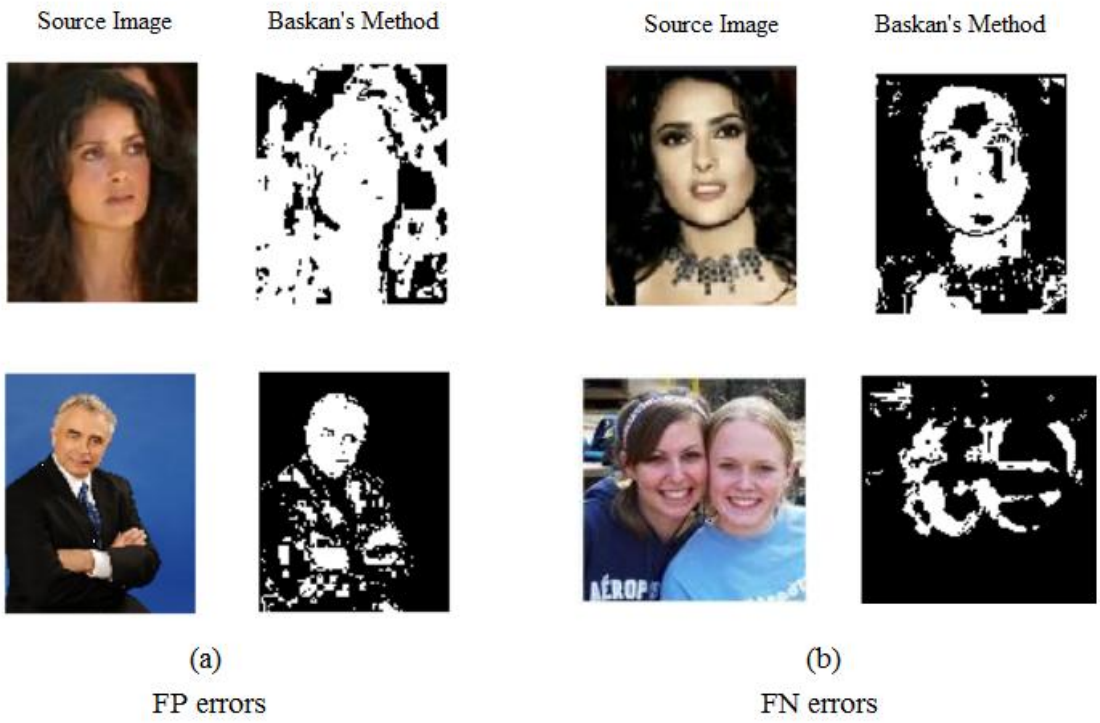


Figure 4.30: Skin detection results using Baskan's Method (2002) applied on real images; (a) examples of FP errors; (b) examples of FN errors.

The quantitative evaluation of Baskan's Method using our training data is shown in Table 4.3. As shown in this table, the accuracy rate of this method is 84.760% with recall of 80.645% and high FNR of 19.355%. As mentioned in the preceding sections, FN errors is more critical problem (see Section 4.4.1)

Table 4.3: Pixel-based quantitative results of Baskan's method using training data.

No. of Pixels	TP	TN	FP	FN	FNR %	FPR %	ACCURACY %	RECALL %
24,328,670	7,896,645	12,724,351	1,812,487	1,895,187	19.355	12.468	84.760	80.645

4.10.4.4 Garcia & Tziritas (1999) Method - Explicit Thresholds using HSV Model

Garcia *et al.*(1999) reported the following rules in the HSV color space for defining six bounding planes that have been found by successive adjustments according to segmentation results:

$$
\begin{aligned}
&S \geq 10;\ V \geq 40;\ \ S \leq -H - 0.1V + 110\ ; \\
&H \leq -0.4\,V + 75 \text{ and} \\
&\text{If } H \geq 0 \\
&\qquad S \leq 0.08(100 - V)\,H + 0.5\,V; \\
&\text{else} \\
&\qquad S \leq 0.5\,H + 35
\end{aligned}
\tag{4.6}
$$

The authors also used YCbCr color space to define another skin color model:

$$
\begin{aligned}
&\text{If } (Y > 128) \\
&\qquad \theta_1 = -2 + (256 - Y)/16; \\
&\qquad \theta_2 = 20 - (256 - Y)/16; \\
&\qquad \theta_3 = 6; \\
&\theta_4 = -8; \\
&\text{else} \\
&\qquad \theta_1 = 6; \\
&\qquad \theta_2 = 12; \\
&\qquad \theta_3 = 2 + Y/32; \\
&\qquad \theta_4 = -16 + Y/16; \\
&\text{end;}
\end{aligned}
\tag{4.7}
$$

$$
\begin{aligned}
&Cr >= -2*(C \\
&b+24)\ ;\ \ Cr >= -(Cb+17)\ ;\ \ Cr >= -4*(Cb+32)\ \ ; \\
&Cr >= 2.5*(Cb+\theta_1)\ ;\ \ Cr >= \theta_3\ ;\ \ Cr >= 2.5*(\theta_4 - Cb)\ \ ; \\
&Cr <= (220 - Cb)/6\ \ ;\ \ \ Cr <= 4/3*(\theta_2 - Cb)
\end{aligned}
$$

To evaluate these bounding planes of the both skin models, the proposed standard set of test images are used for this purpose. The results of image segmentation are shown in Figure 4.31. In this figure, the left column shows the original image; the second column shows skin segmentation output using HSV color space; the third column shows skin segmentation output using YCbCr color space. As shown in this figure, the skin model using YCbCr color space shows no skin regions at all for all test images. We test the rules again manually for some test pixels according to the drawing shown by the authors. These rules also failed manually. The failure of these rules to detect skin regions may return to printing mistakes in the original paper.

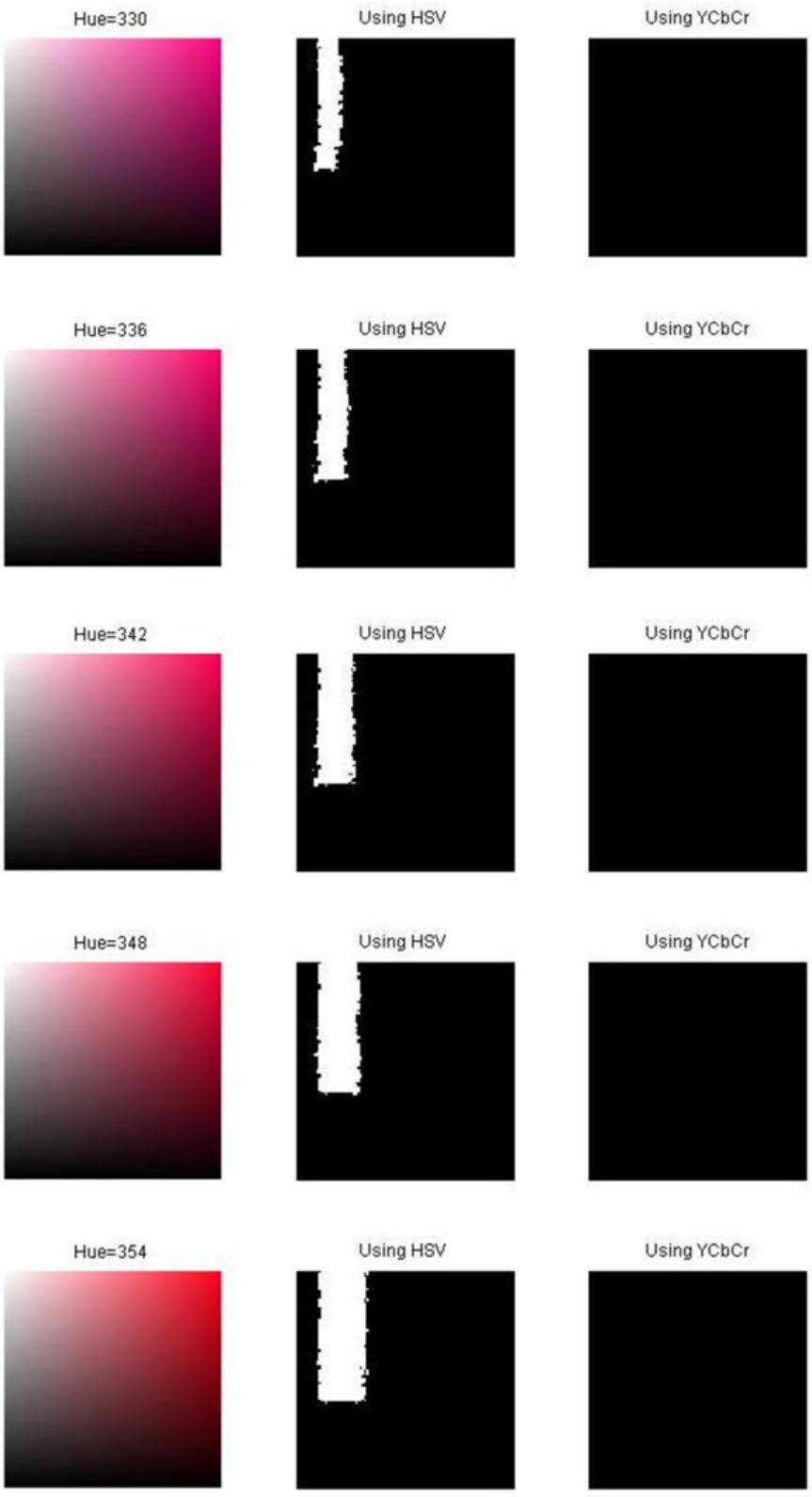


Figure 4.31: Skin detection results using Garcia *et al.* (1999) Approach.
The left column shows the original image; second column shows skin segmentation output using HSV color space. The output of YCbCr color space is shown in the third column.

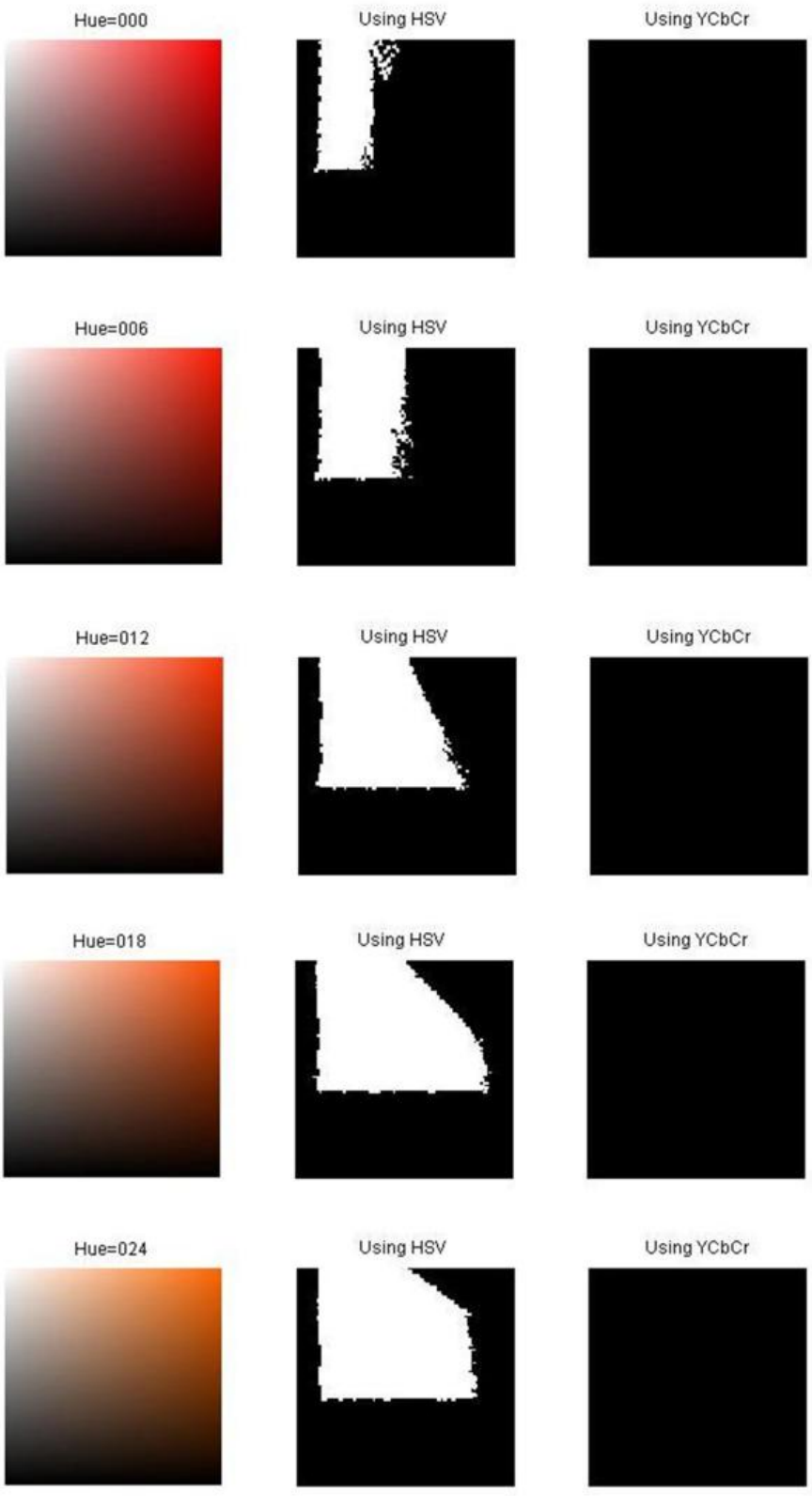


Figure 4.31(*Continued)*: Skin detection results using Garcia *et al.* (1999) Approach. The left column shows the original image; second column shows skin segmentation output using HSV color space. The output of YCbCr color space is shown in the third column.

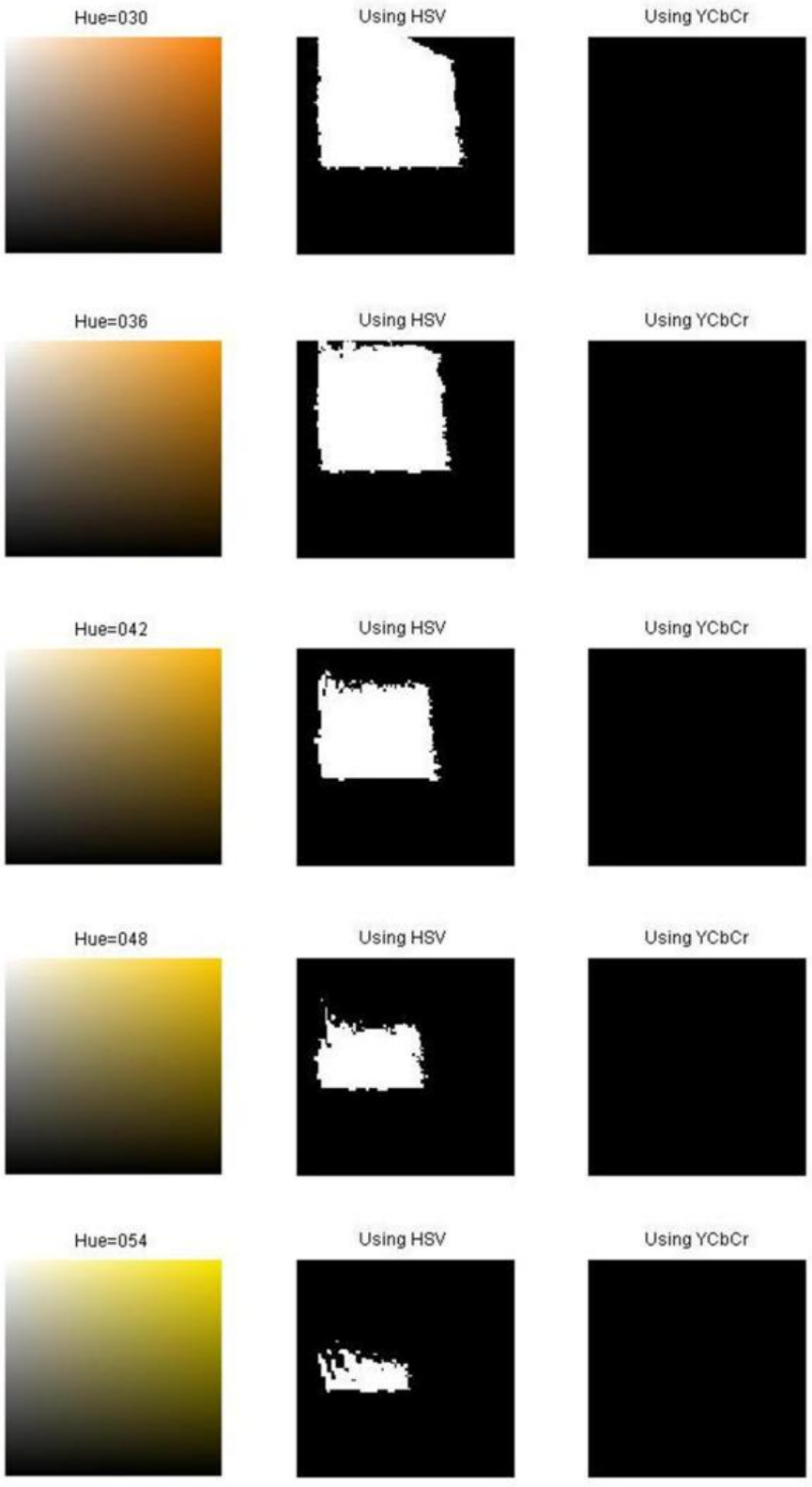


Figure 4.31 (*Continued*): Skin detection results using Garcia *et al.* (1999) Approach. The left column shows the original image; second column shows skin segmentation output using HSV color space. The output of YCbCr color space is shown in the third column.

The feasibility of the classification boundary using HSV color space is acceptable although they are not perfect. For instance, Figure 4.32 shows examples of FN errors using Garcia's method (1999) applied on test images. The Left column shows the original image; second column shows skin segmentation output. The figure shows three examples of FN. In each row, the pixel marked ? evidently belongs to skin class, but the classification boundaries incorrectly classified it as non-skin.

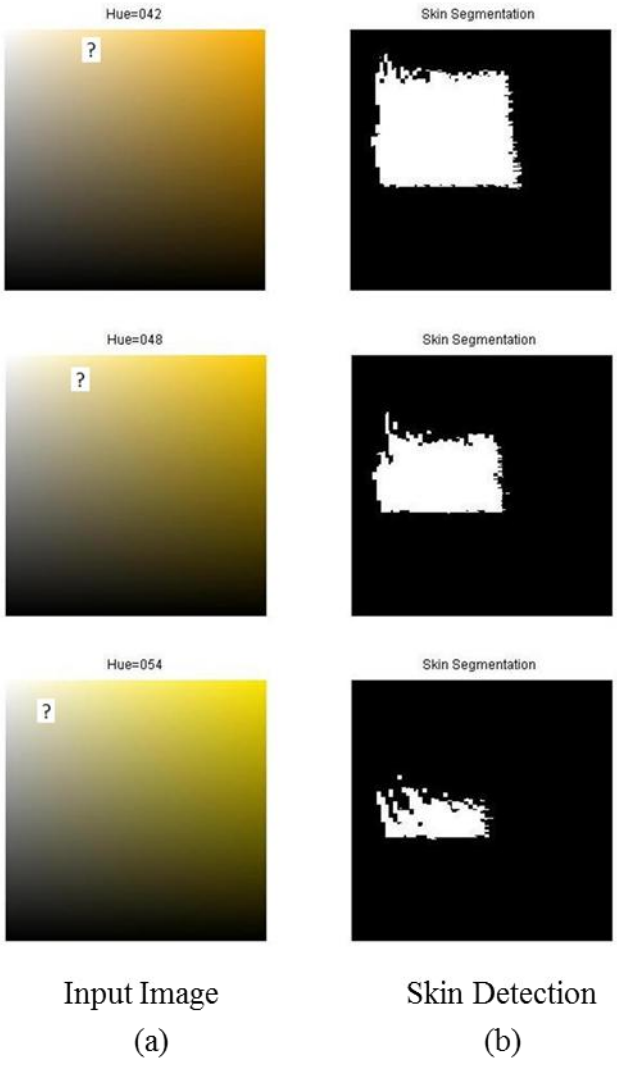


Figure 4.32: Skin detection results using Garcia *et al.* (1999) approach applied on test images Hue=42, 48, and 54. (a) Input image; (b) skin detection output. The figure shows three examples of FN; the pixel marked ? evidently belongs to skin class, but the classification boundaries incorrectly classified it as non-skin.

An example of the qualitative evaluation of Garcia's method using real images is shown in Figure 4.33. As shown in this figure, this method shows high FN errors in which a large number of human skin pixels are missed (i.e. wrongly classified as non-skin pixels).

Figure 4.33: Examples of FN errors using Garcia's method applied on real images.

The quantitative evaluation of Garcia's method using our training data is shown in Table 4.4. As shown in this table, the accuracy rate of this method is 88.055% with recall of 71.039% and high FNR of 28.961%. As mentioned in the preceding sections FN errors is more critical problem (see Section 4.4.1).

Table 4.4: Pixel-based quantitative results of Garcia's method using our training data.

No. of Pixels	TP	TN	FP	FN	FNR %	FPR %	ACCURACY %	RECALL %
24,328,670	6,956,050	14,466,457	70,381	2,835,782	28.961	0.484	88.055	71.039

4.10.4.5 Bayes Classifier based on Uni-Skin Model

Naive Bayes Classifier is used to estimate skin color distribution from the training data without deriving an explicit model of the skin color. Its goal is to drive classification boundaries which are optimal in the sense that its use minimizes the probability of classification errors.

In this method, the training data is divided into two classes (that are skin and non-skin samples) using HSV color space. In this approach, the full color components (i.e. H, S and V) are used to estimate the probability distribution. The Bayes classifier classifies data in two steps:

- Training step: by using the training data (samples), the method estimates the parameters of a probability distribution.
- Prediction step: for any unseen test sample, the method computes the posterior probability of that sample belonging to each class. The method then classifies the test sample according to the largest probability

Examples of the classification boundaries of this method using the proposed standard set of test images are as shown in Figure 4.34. The first column shows the original test images. The second column shows the real distribution of raw data. The classification boundaries of Bayes classifier are shown in the third column. As shown in this figure, the classification boundaries did not correspond to the real distribution of the raw data although the classifiers minimize the total amount of error. This problem comes from the fact that non-skin samples are located in two regions that are in the opposite sides of the image (i.e. the upper right corner and lower left corner). Since the classifiers treat these non-skin samples as one class, the classification boundaries would be affected accordingly (i.e. shifted). As will see in next sections, this problem can be solved by treating the non-skin samples as two classes rather than one class.

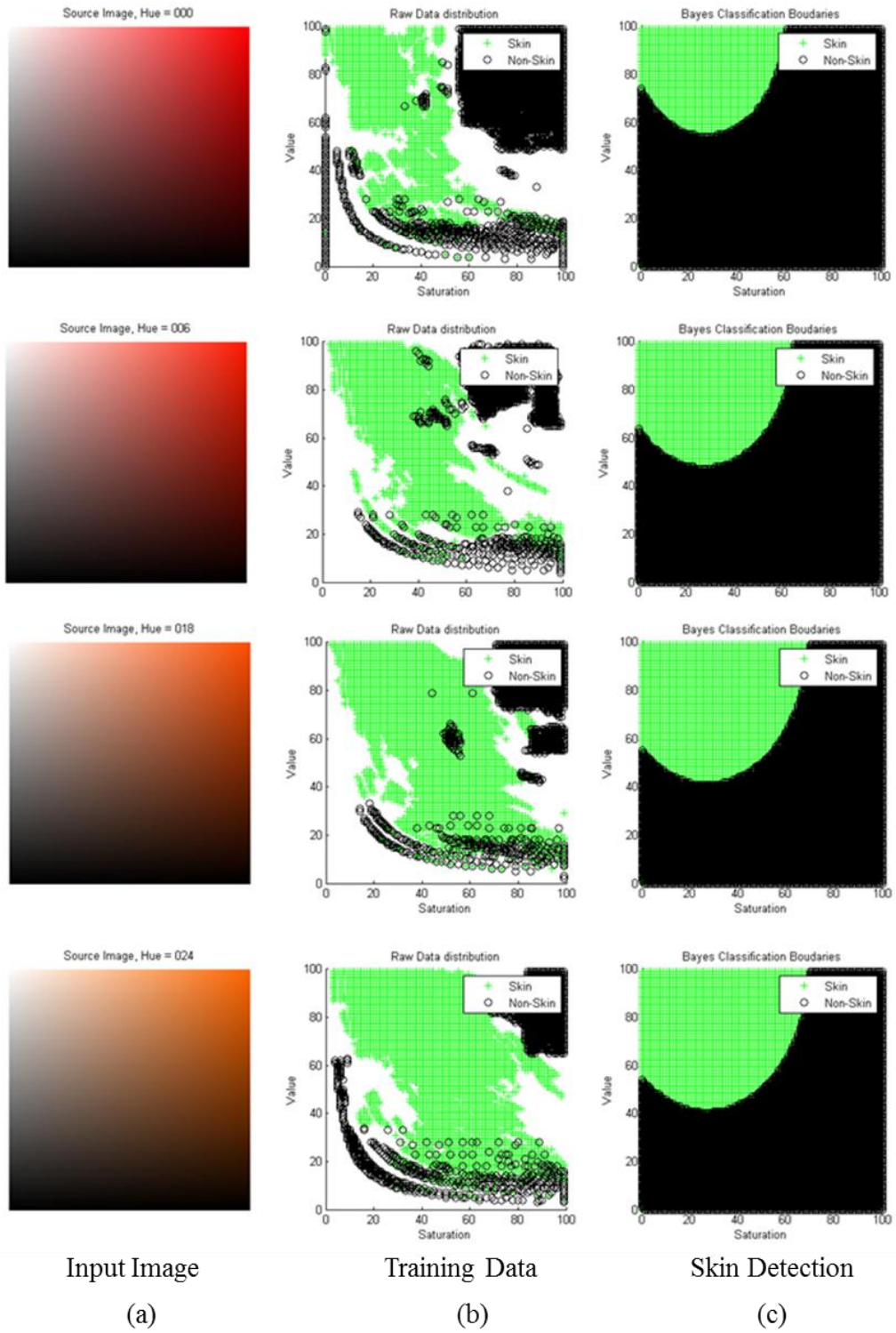


Figure 4.34: Skin detection results using Bayes classifier based on two-class classification problem. (a) input image; (b) training data distribution; (c) skin detection results.

The pixel-based quantitative evaluation of this method using our training data is shown in Table 4.5. This method shows acceptable classification accuracy with detection accuracy of 93.882% and recall of 86.776%. Although the FPR is low 1.331%, the FNR is high 13.224%. We aim at better accuracy results.

Table 4.5: Pixel-based quantitative results of Bayes classifier based on two-class classification problem using our training data.

No. of Pixels	TP	TN	FP	FN	FNR %	FPR %	ACCURACY %	RECALL %
24,328,670	8,496,916	14,343,413	193,425	1,294,916	13.224	1.331	93.882	86.776

4.10.4.6 Bayes Classifier based on Multi-skin Models

Since the non-skin samples are clustering in two different regions (i.e. the upper right UR corner and the lower left LL corner), we propose to treat non-skin samples as two classes in order to get more feasible results. Later, the results of these two-classes are combined into one group because they characterize the background. As mentioned before that we already proposed multi-skin models in this research. Therefore, we have six-class classification problem (i.e. four skin classes and two non-skin classes) as shown in Figure 4.12 (Section 4.6).

The proposed set of test images is used for inspecting the feasibility of the classification boundaries and the results are shown in Figure 4.35. For abstraction, only six test images are shown here. More test images can be found in APPENDIX-C. As shown in this figure, each test image is shown along with three graphs; these are: raw data distribution graph, the classification boundaries as two-class classification problem (i.e. skin and non-skin), and the classification boundaries using multi-skin modeling (i.e. 4-skin classes graph). The skin and non-skin graph shall be used to evaluate the method. Recognizing this, we combined the 4-skin classes together into one class and then the two non-skin classes are also combined into one class. So, we have 2-class classification problem. The 4-skin classes graph shows the classification boundaries of different skin classes.

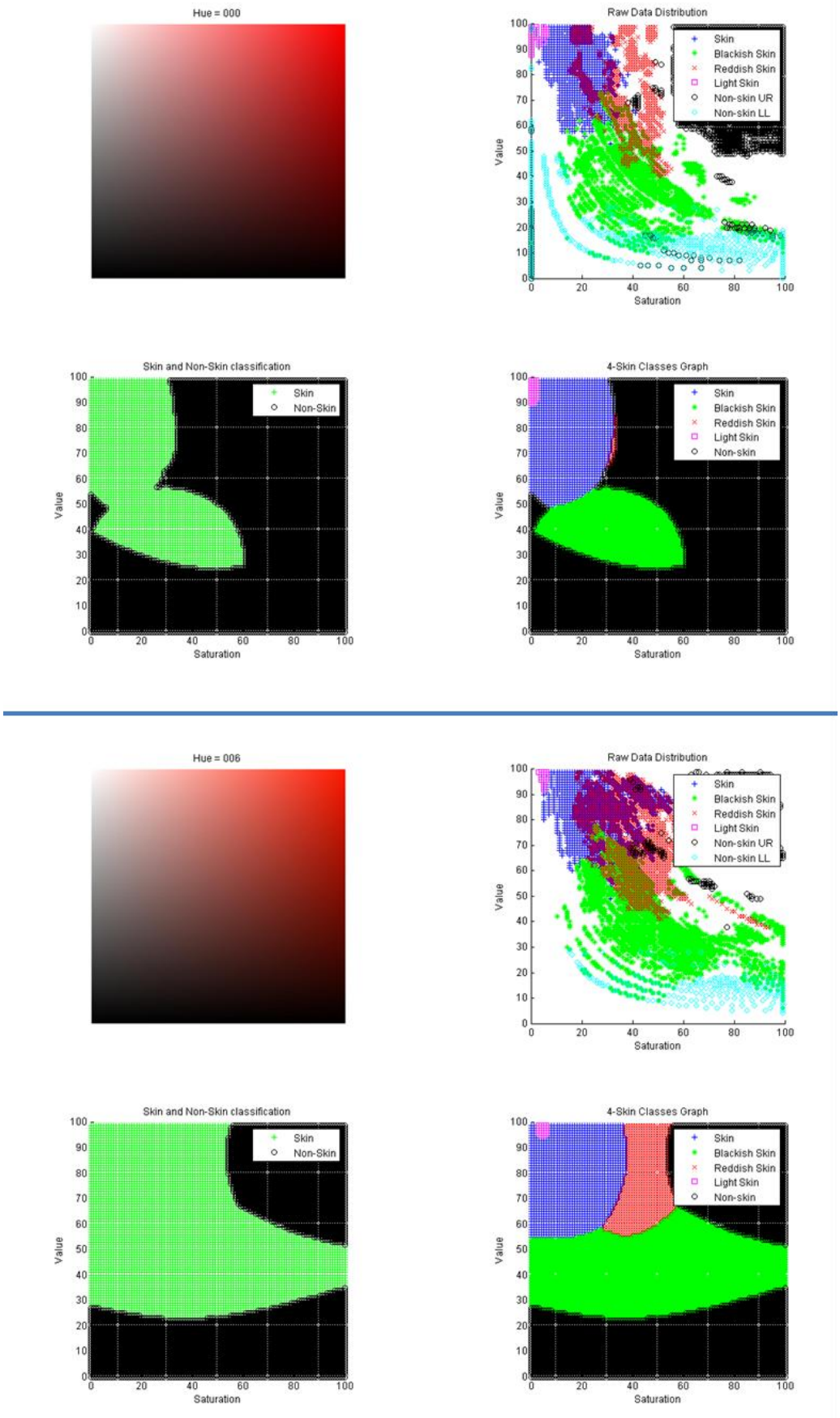


Figure 4.35: Skin detection results using Bayes classifier based on multi-models applied on standard set of test images; hue=00 and hue=06.

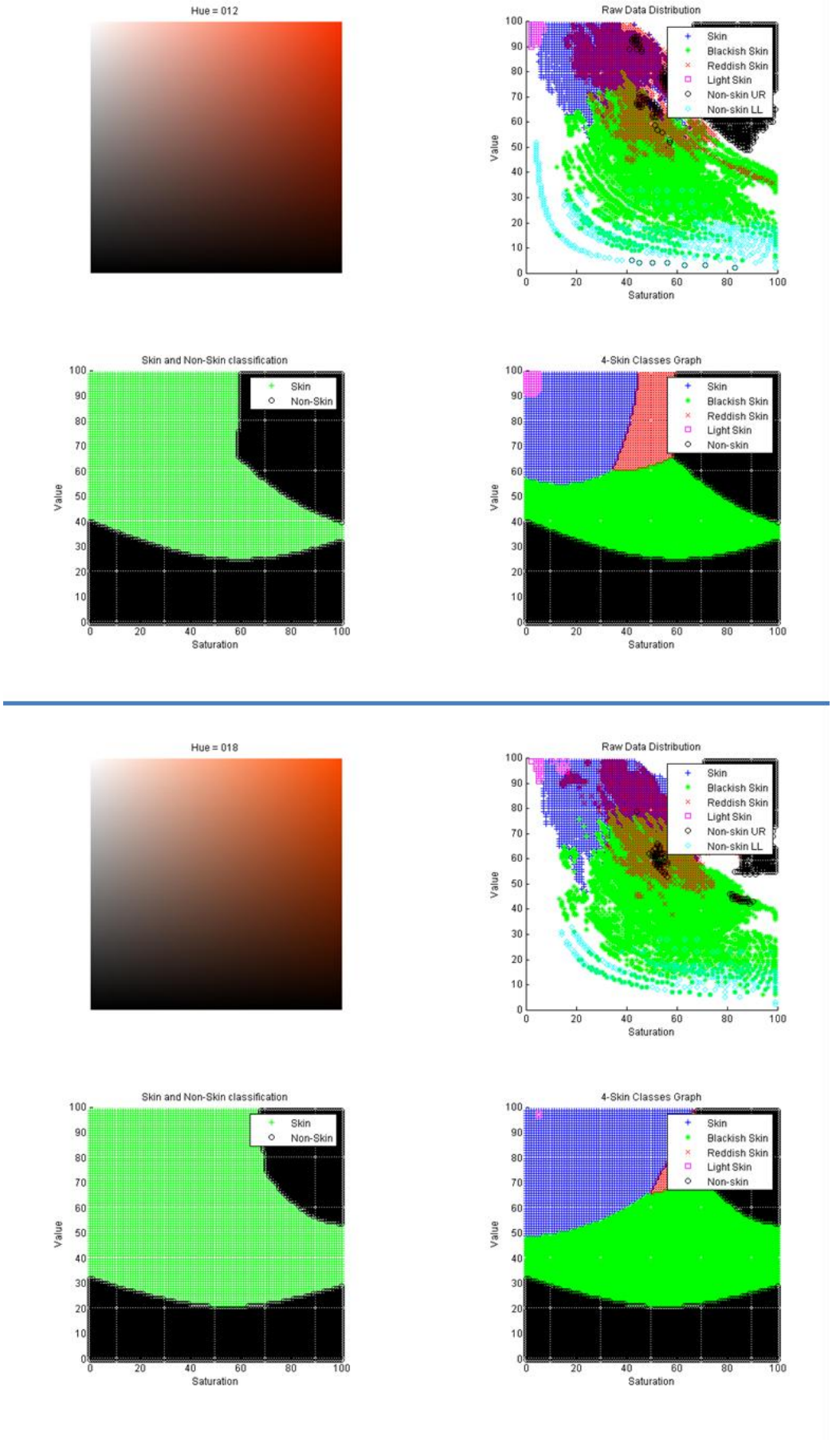


Figure 4.35(*Continued)*: Skin detection results using Bayes classifier based on multi-models applied on standard test images; hue=12 and hue=18.

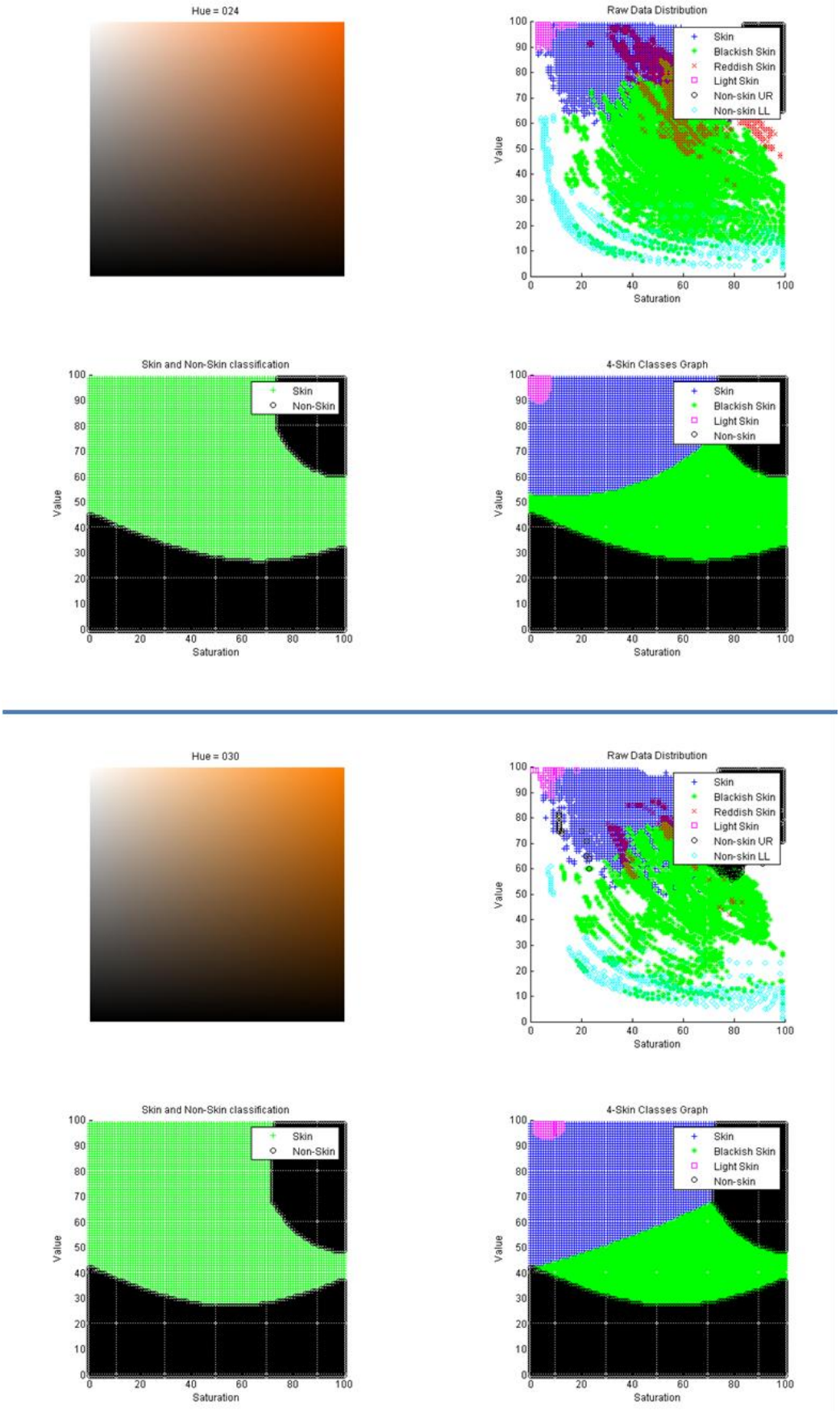


Figure 4.35(*Continued)*: Skin detection results using Bayes classifier based on multi-models applied on standard test images; hue=24 and hue=30.

The feasibility evaluation of the classification boundaries using this method shows many drawbacks in many cases such as:

- Test image at hue=000: The classification boundaries did not correspond to the real distribution of raw data. Furthermore, the reddish-skin region m_3 (i.e. labeled as Red Cross sign) is completely eliminated. On the other side, the existence of gulfs is not preferable.
- Test image at hue=06: The region that corresponds to light-skin class m4 (i.e. labeled as pink square sign) shows that the region forms a gulf while its location at hue=0 and hue=12 images is logically acceptable (these images come before and after the mentioned image). Furthermore, compared to image at hue=00, one can notice that there is a big difference between them. Abrupt change in the shape through successive adjacent slides is undesirable.
- Test image at hue=24: The segmentation shows that the reddish skin region m_3 (i.e. labeled as red cross) is completely eliminated.
- Test image at hue=36 (shown in APPENDIX-B): The segmentation of image shows that the contour of skin classes, in general, is irregular.
- Test image at hue=42 (shown in APPENDIX-B): The segmented region of the blackish skin class ω2 (i.e. labeled as green star) is so small, although it exists normally in both images that come before and after this test image (i.e. test images at hue= 42 and 54).

Table 4.6 shows the quantitative results of this method computed for each hue slide separately using our raw data. The general performance of this method using raw data is shown in Table 4.7. In this table, the Bayes classifier using multi-skin models shows classification accuracy of 97.267% and recall of 95.609%. The FNR and FPR are 4.391% and 1.615% respectively. This method show reasonably good predictive accuracy but we aim at better accuracy results.

Table 4.6: Pixel-based quantitative results of Bayes classifier using multi-skin color models.

Seq.	Hue	No. of Pixels	TP	TN	FP	FN	FNR %	FPR %	ACCURACY %	RECALL %
1	0	2,050,641	356,899	1,653,554	25,088	15,100	4.059	1.495	98.040	95.941
2	6	806,618	543,212	248,183	5,486	9,737	1.761	2.163	98.113	98.239
3	12	1,838,248	1,087,716	714,743	14,759	21,030	1.897	2.023	98.053	98.103
4	18	3,145,858	2,835,059	255,864	11	54,924	1.900	0.004	98.254	98.100
5	24	3,296,003	2,747,600	487,360	4,586	56,457	2.013	0.932	98.148	97.987
6	30	1,441,690	1,142,011	263,923	1,368	34,388	2.923	0.516	97.520	97.077
7	36	590,397	220,067	356,205	6,280	7,845	3.442	1.732	97.608	96.558
8	42	1,033,412	123,123	894,438	12,036	3,815	3.005	1.328	98.466	96.995
9	48	771,502	107,609	641,417	14,905	7,571	6.573	2.271	97.087	93.427
10	54	566,411	32,525	503,915	29,971	-	0.000	5.614	94.709	100.00
11	60	859,638	40,832	717,939	100,867	-	0.000	12.319	88.266	100.00
12	330	714,675	33,592	674,813	3	6,267	15.723	0.000	99.123	84.277
13	336	2,285,363	7,821	2,272,220	6	5,316	40.466	0.000	99.767	59.534
14	342	2,223,753	5,348	2,200,538	2	17,865	76.961	0.000	99.197	23.039
15	348	1,578,308	16,314	1,499,944	6	62,044	79.180	0.000	96.069	20.820
16	354	826,478	27,474	672,442	8	126,554	82.16	0.00	84.69	17.84

Table 4.7: The general performance of Bayes method using multi-skin models.

Seq.	Hue	No. of Pixels	TP	TN	FP	FN	FNR %	FPR %	ACCURACY %	RECALL %
1	ALL	24,328,670	9,361,881	14,302,008	234,830	429,951	4.391	1.615	97.267	95.609

4.10.4.7 Linear Discriminant Analysis (LDA)

Linear discriminant analysis (LDA) is a classic method of classification in statistics, pattern recognition, and machine learning to find a linear combination of features which separates two or more classes of objects. To use discriminate analysis in this study, the collected data of skin and non-skin samples which are divided into six classes, are first considered.

To evaluate the feasibility of the LDA classification boundaries, the proposed standard set of test images is used and the results are shown in Figure 4.36. For abstraction, only six test images are shown here. More test images can be found in APPENDIX-D. As shown in this figure, each test image is shown along with three graphs; these are: raw data distribution graph, the classification boundaries as two-class classification problem (i.e. skin and non-skin), and the classification boundaries using multi-skin modeling (i.e. 4-skin classes graph).

As shown in Figure 4.36, the feasibility evaluation of segmentation output using this method shows many drawbacks in many cases and the decision boundaries clearly don't reflect the actual distribution of training data. For example, the visual inspection of classification boundaries at test image hue=006 shows that the white-skin class (i.e. labeled as blue plus sign) tends to have very small narrow region compared to its actual large distribution that is shown in the raw data distribution graph. In general, the classification boundaries of all test images do not match the real distribution of raw data. Furthermore, abrupt change in the shape through successive adjacent slides is undesirable.

Table 4.8 shows the quantitative results of this method computed for each hue slide separately using our raw data. The general performance of this method using raw data is shown in Table 4.9. As shown in this table, the LDA classifier shows classification accuracy of 71.279 % and recall of 82.355 %. The FNR and FPR are 17.645 % and 36.293% respectively. It is clear that the results are poor compared to the previously described methods. For instance, the accuracy rate at hue=06 is 28.419% with a recall of 16.883%.

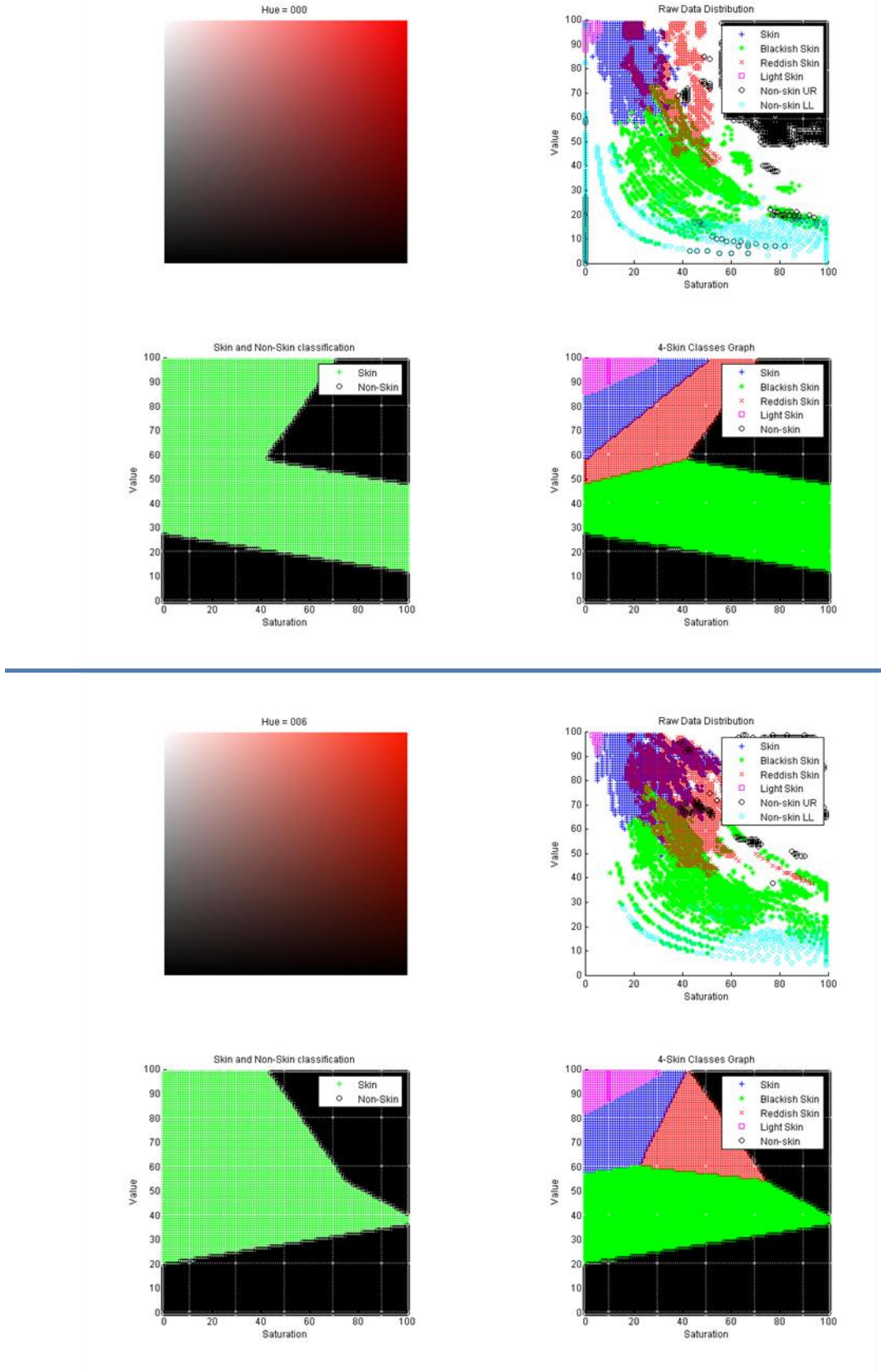


Figure 4.36: Skin detection results using LDA Classifier.

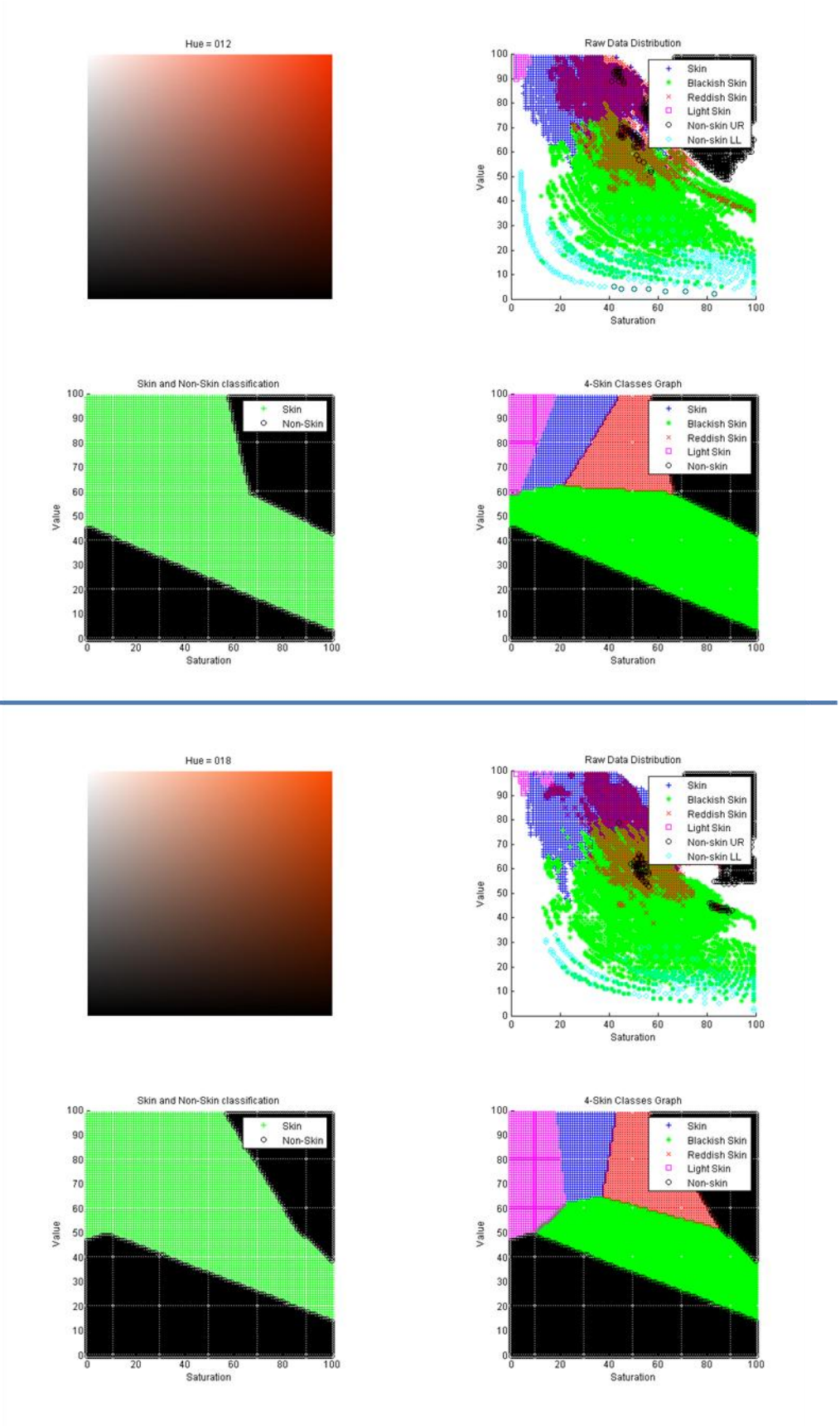


Figure 4.36 *(Continued):* Skin detection results using LDA Classifier.

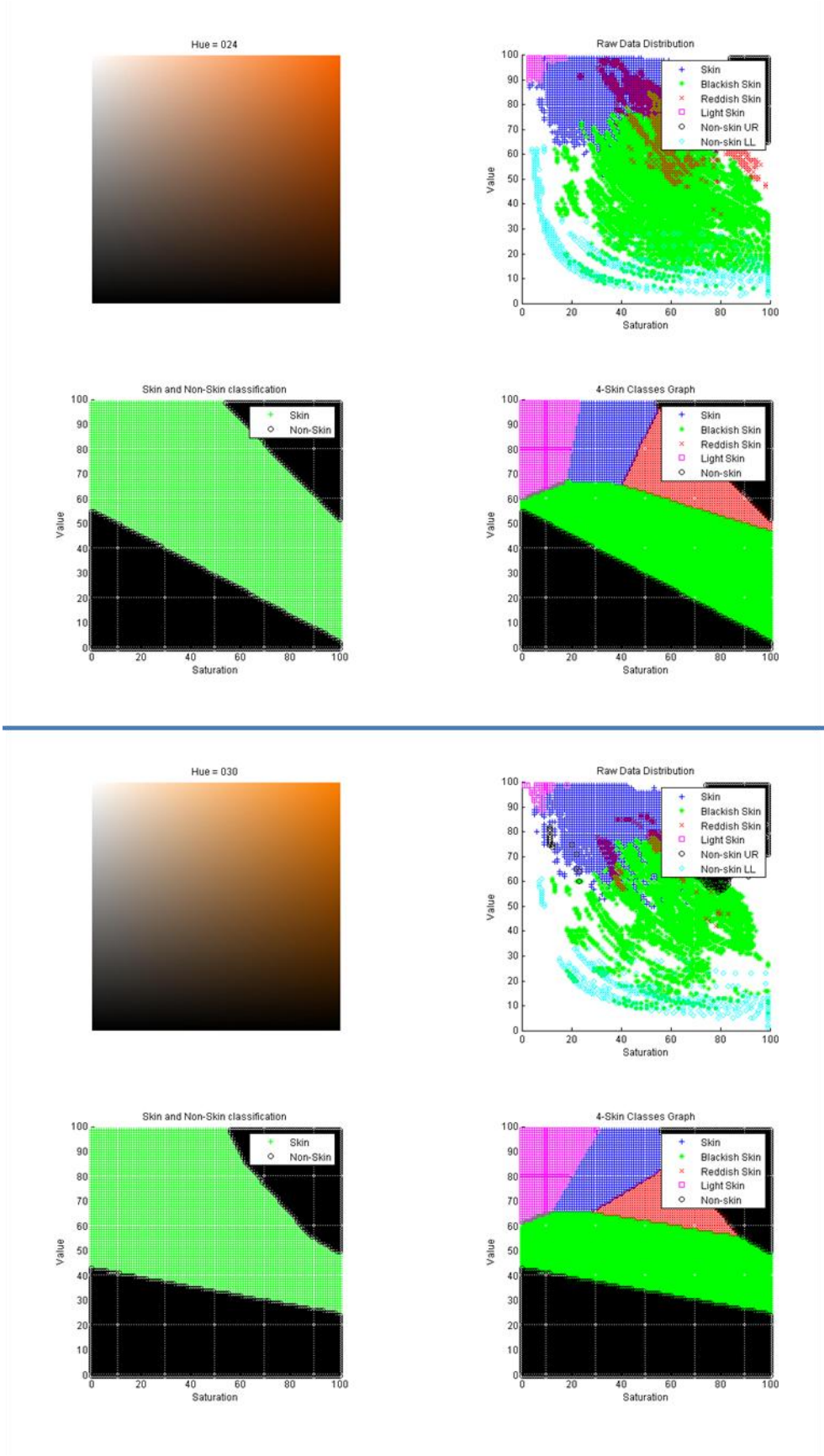


Figure 4.36*(Continue):* Skin detection results using LDA Classifier.

Table 4.8: Pixel-based quantitative results of LDA Classifier using multi-skin models.

Seq.	Hue	No. of Pixels	TP	TN	FP	FN	FNR %	FPR %	ACCURACY %	RECALL %
1	0	2,050,641	279,344	1,295,857	382,785	92,655	24.907	22.803	76.815	75.093
2	6	806,618	93,357	135,877	117,792	459,592	83.117	46.435	28.419	16.883
3	12	1,838,248	999,193	586,431	143,071	109,553	9.881	19.612	86.257	90.119
4	18	3,145,858	2,532,966	215,014	40,861	357,017	12.354	15.969	87.352	87.646
5	24	3,296,003	2,742,953	405,177	86,769	61,104	2.179	17.638	95.514	97.821
6	30	1,441,690	1,050,544	162,715	102,576	125,855	10.698	38.665	84.155	89.302
7	36	590,397	199,707	285,839	76,646	28,205	12.375	21.145	82.241	87.625
8	42	1,033,412	70,297	628,028	278,446	56,641	44.621	30.717	67.575	55.379
9	48	771,502	38,340	510,019	146,303	76,840	66.713	22.291	71.077	33.287
10	54	566,411	7,529	349,237	184,649	24,996	76.852	34.586	62.987	23.148
11	60	859,638	6	662,996	155,810	40,826	99.985	19.029	77.126	0.015
12	330	714,675	20	401,844	272,972	39,839	99.950	40.451	56.230	0.050
13	336	2,285,363	243	1,577,825	694,401	12,894	98.150	30.560	69.051	1.850
14	342	2,223,753	975	1,260,375	940,165	22,238	95.800	42.724	56.722	4.200
15	348	1,578,308	2,813	304,764	1,195,186	75,545	96.410	79.682	19.488	3.590
16	354	826,478	16,384	310,882	361,568	137,644	89.363	53.769	39.598	10.637

Table 4.9: The general performance of LDA classifier using multi-skin models.

Seq.	Hue	No. of Pixels	TP	TN	FP	FN	FNR %	FPR %	ACCURACY %	RECALL %
1	ALL	24,028,995	8,034,671	9,092,880	5,180,000	1,721,444	17.645	36.293	71.279	82.355

4.11 The Proposed Algorithm

As shown in the previous sections, many methods have been proposed to build a skin color model. Although the same training data and same standard set of test images are used, it has been proven that these methods produce different skin models (or classifications boundaries) and each one has its own drawbacks. This research aims at developing a novel method for skin color modeling and detection that overcomes the drawbacks of these methods to improve segmentation accuracy. From the extensive experiments in this research it is found that there are two sub-problems in the existing methods:

- The limitations of uni-skin modeling to cover different skin color tones. In this research, this sub-problem had been already solved by shifting to multi-skin modeling. The details are presented in Section 4.6.

- Data collection problems. The raw data contain many serious issues and sub-problems that should be considered before going to the training and classification stage. Many researchers deal with the collected data as true samples (Chai et al., 2003; Chaves-González et al., 2010; Chen & Wang, 2007; Frisch et al., 2007; Garcia & Tziritas, 1999; Jones & Rehg, 2002; Moallem et al., 2011; Phung et al., 2005; Sandeep & Rajagopalan, 2002; Shih et al., 2008; Terrillon et al., 2000; Tomaz et al., 2004; Yuetao & Nana, 2011; Zaqout et al., 2004). In this research we found that these issues are highly important and have a direct effect on the shape of samples distribution and consequently the classification boundaries. If substantial variations in in-class properties are presented, the resulting segmentation may have many regions which are misclassified.

The remainder of this section illustrates the main issues implied in raw data and then our proposed step-by-step algorithm.

4.11.1 Issues of Raw Data and Sub-Problems

In this section we will discuss some issues of raw data and then our proposed solutions to deal with these issues.

i) **Data richness:** Although more than 20,000,000 training pixels are collected for this research, not all colors are encountered. As a result, many small gaps (or holes) appear in the histograms. The zeros entries mean that many instances of color-tones are not present in the training samples. This is normal in many other applications where the missing samples (i.e. unseen new patterns) are predicted from training data. The small gaps in the chart can be filled in a variety of ways. Theoretically, as more data samples are used, more regions in the sub-space become filled. Selecting more pixels may help to make a good estimation of the color distribution of skin, but non-skin pixels could be erroneously included too. Selecting fewer pixels can minimize the probability of including non-skin pixels, but the estimated skin color distribution could not match well with the real situation. It was noticed, however from experience, that more samples cause data redundancy, increase the overlapping between classes, and more noise. This is a typical trade-off issue. As RGB, HSV, etc. color spaces (i.e. 24 bits) offer about 16.7 million distinct colors, we found that this solution is not practical. Jones and Rehg (2002) built a 3D RGB histogram model with one billion pixels collected from 18,696 web images. They reported that 77% of the possible RGB colors are not encountered and most of the histogram is empty.

ii) **Double-classes and overlapping:** double-class means that the same training pixel comes from more than one class which makes the problem harder. With various image types and sources, this usually happens during data collection when a training pixel is assigned to a class in an arbitrary image, while the same color pixel appears again in another image and assigned to another class. Humans do not deal with colors as numbers like computers. This causes the same training pixel to be double-classes (i.e. data collection errors). Jones and Rehg (2002) developed one of the most comprehensive accounts on skin color models for

uncontrolled still images. Their study shows a significant overlap of skin and non-skin models, which limits the performance of the skin detector (Jones & Rehg, 2002).

iii) **Invalid data:** when raw data samples (i.e. skin patches) are collected manually from arbitrary images, many anti-facts such as small spots, lumps, specks and freckles may be found in these patches where there should be none. These anti-facts are invalid data or noise. For example, patches of non-skin samples are cropped manually from the background (e.g. curtains, couch, buildings, etc.). These non-skin samples may contain many pixels which are skin-like in color. In such case, if the frequency of such colors does not appear again in skin training samples, it dramatically affects the contour of the classification boundaries. Invalid data degrades the system accuracy because it will be difficult to determine which version of the data is correct. In most color spaces, the adjacent entries show very similar colors; therefore it is inconceivable that while some regions in a small neighborhood belong to the skin, the other regions that fall in between do not. For instance, let us consider again the example shown in Figure 4.18 and its new version shown in Figure 4.37. Although, the authors discussed this situation as true data, this example shows the effect of invalid data on the shape of classification boundaries. The question is: "How can we get pure and true samples?". In this research we insist on dealing with such cases as noise. Therefore, instead of searching for complex classification boundaries, we propose to apply pre-processing steps for noise removal, gulfs removal, and boundary smoothing. This will help seek simpler and more accurate boundaries. Figure 4.37 shows an example for correcting noise data inspired by morphological operations. In this figure, the noisy pixel had assigned new class label (that is class 2) driven from local neighborhood class. So, the contour of classification boundaries will be changed accordingly. In other words, the classification boundaries can be simplified, motivated by the idea that the underlying skin models will not require decision boundaries that are complex due to noise.

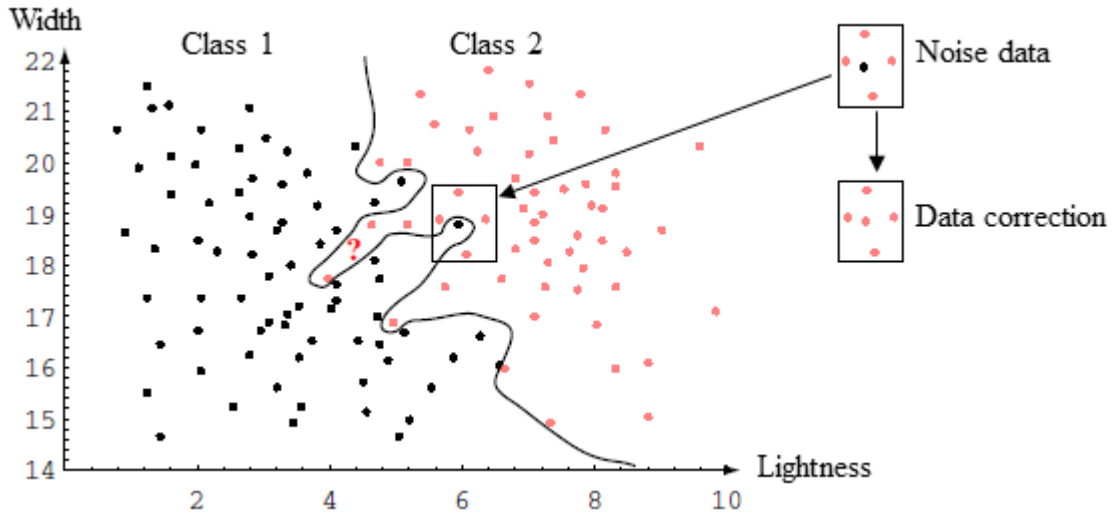


Figure 4.37: Data correction and noise removal.

iv) **Data generalization:** machine leaning and statistical pattern recognition methods depend mainly on the training data to find the decision functions. If the skin samples are collected from images of a specific database, then the results depend mainly on testing images of the same database. If test images from another database are used these functions may fail. The question is; "How can we generalize the raw data?".

For any proposed classification algorithm, there is a trade-off between the above-mentioned issues and the accuracy of the classification boundaries yielded. For example, noise reduction improves the quality of data and then the accuracy of classification boundaries. Consequently better segmentation results are obtained. Therefore, pre-processing steps for improving the quality of raw data are required.

In this research, we proposed to use the idea of dominant-class filter for sub-sampling the 100×100 SV-plane into 20×20 plane. The filter is passed over the SV-plane. It receives 5×5 region and outputs the dominant-class as shown in Figure 4.38. Dominant-class denotes the values shared by the majority of the pixels in a given region. This step is used for noise removal as well as to predict the class of unseen pixels based on the pixel's neighborhood class. In

Figure 4.38 it is clear that the class of value 5 is the dominant class while there is missing data (i.e. entries of value zero) and noise (i.e. class value 3).

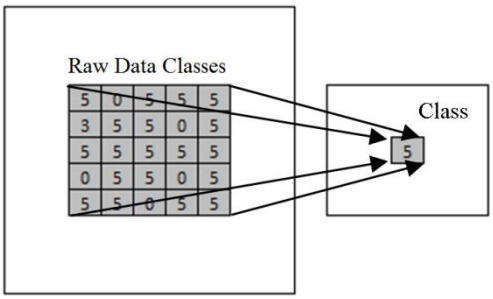


Figure 4.38: The dominant-class (majority) filter is useful for filling holes and noise removal. The filter receives 5×5 region and outputs the dominant-class.

Then, morphological operations are used to remove holes, gulfs, and protrusions in order to create compact regions. Morphological operations can be described simply in terms of adding or removing pixels from a region according to certain rules, which depend on the pattern of neighboring pixels. Erosion removes pixels from a region in an image or, equivalently, turns OFF pixels that were originally ON. The purpose is to remove pixels that should not be there (i.e., in our case, noise data). Dilation can be used to add pixels to a region.

The first row of Figure 4.39 shows an example of raw data distribution containing missing data and noise that form holes. The figure describes the steps (from left to right) for filling holes. The second row shows an example of removing thin gulfs (from left to right).

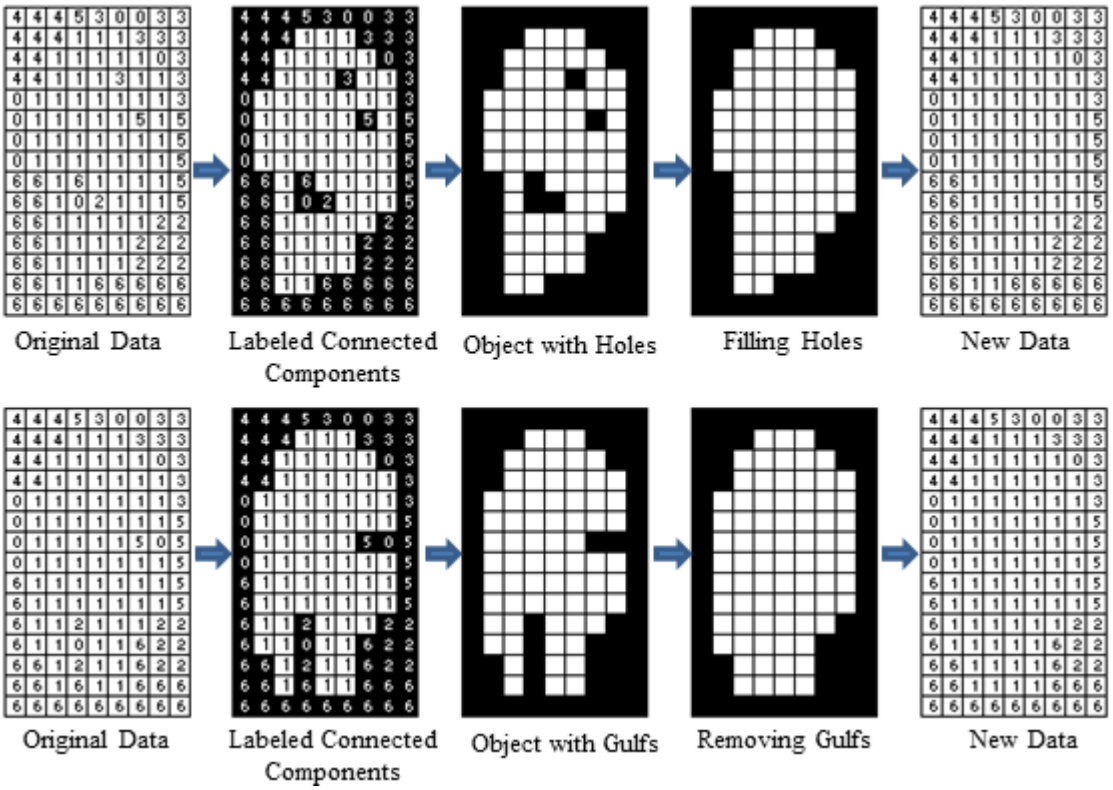


Figure 4.39: Noise removal based on morphological operations; the first row shows an example of filling holes; second row shows an example of removing thin gulfs.

4.11.2 Step-by-step Algorithm

The step-by-step algorithm for determining the classification boundaries is listed as follows:

Step 1. The goal of this step is to produce tabular summaries. This step is also useful for removing double-classes problem as follows:

- A separate 3D histogram for each class is constructed by counting the relative frequency of each pixel value that occurs in the training samples (i.e. frequency) as in Figure 4.40(a). A 3D matrix of size 100×100×60 bins is used to represent each histogram (i.e., 100×100 SV-plane × 60 hue slides). The histogram counts the number of occurrences of a colored pixel x using its color information (i.e. H, S, and V), where $f(x)$ gives the frequency in the histogram entry associated with the HSV color triple.
- All histograms are merged into one tabular summary called the Class-Labeled Summary Table 3D-CLST (size of 100×100×60 bins) as in Figure 4.40(b). This step is used to resolve

the ambiguity due to double-classes and overlapping. In real raw data, the frequency of a colored pixel x may appear in more than one class (i.e., double-classes are attributed to data collection errors and noise). Based on a simple heuristic, when the frequency of a colored pixel x is the maximum at class m_i then it can be said that x comes from class m_i. Therefore, pixel x is assigned to class m_i, and its corresponding entry at the CLST is marked with class label m_i.

Step 2. The goal of this step is to augment the raw data using the pixel's neighborhood operations. The idea is that the class of missing samples (i.e., pixels not yet seen) could be predicted from the neighbored pixels. This step is based on a simple heuristic that the adjacent entries in the color space show very similar colors and their spatial variation is gradual. In other words, the pixel's new value must be computed from the values of pixels in its vicinity. This process is performed in two steps:

- Initially, the 3D-CLST is projected onto a set of 2D SV-planes. Figure 4.40(c) illustrates an example of the SV-plane. In this figure, each colored pixel describes the class label m_i that it comes from. Entries of value zero mean that the pixels are not assigned to any class (i.e., pixels not yet seen).
- Each SV-plane in the color space is divided into 400 samples, that is, 20×20 patches, as in Figure 4.40(d) which maintain a fair accuracy in the system.
- The idea of the dominant class is used to predict the class of noise data or missing samples based on the values of pixels in its vicinity. First, a filter window of size 5×5 pixels is used to reduce the size of the 100×100 SV-plane into a 20×20 plane (i.e., sub-sampling). The filter receives as input a 5×5 pixel region and generates an output value that represents the dominant-class in that region. Figure 4.40(e) shows the output of the sub-sampling step with dominant-class output.

Step 3. The goal of this step is to produce compact regions. In this research we proposed the use of erosion and dilation operations inspired from morphological operations in binary images. The

newly-generated 20×20 plane is considered as a binary image containing label-connected components. The group of connected pixels (i.e., pixels that have the same class label) form a region (object) in that image, and the label of the pixels refers to the region's number. For example, the pixels labeled 1 make up region one; the pixels labeled 2 make up a second region, and so on. Generally, the scattered plot of raw data forms non-compact regions (with thin gulfs, protrusions, and holes). This steps aims at filling holes and removing small protrusions and thin gulfs. For this purpose, structuring elements of different sizes are passed over each plane to create compact regions. Figure 6(f) shows examples of removing thin gulfs, filling holes, and smoothing the region's border. The idea of this step was already shown in Figure 4.39

Step 4. The newly-generated region boundaries are compared with the previous situation. If they are matched (i.e., no change), the 20×20 SV-plane is resized back into a 100×100 SV-plane, and the contents of the SV-plane are saved as the new representative training data, instead of the old raw data. Otherwise, the region boundaries are updated and Step 3 is repeated.

Step 5. Reconstruct the 3D histogram using newly generated class-labeled regions and pixels color frequencies and then pass them to Bayes classifier to find new regional boundaries.

Step 6. Depending on human perception and the specific user's judgment, the boundaries are adjusted to show the best results experimentally.

Step 7. The classification boundaries are transformed into three-dimensional Lookup-Table to speed up the system. Each SD-LUT cell contains information about the classification result of any HSV color. The size of the SD-LUT is (100×100×60) cells and it is indexed by a color information vector (H, S, and V).

As mentioned before, the SD-LUT is stored in the system secondary storage. When the skin detector program is initialized, it just reloads the SD-LUT.

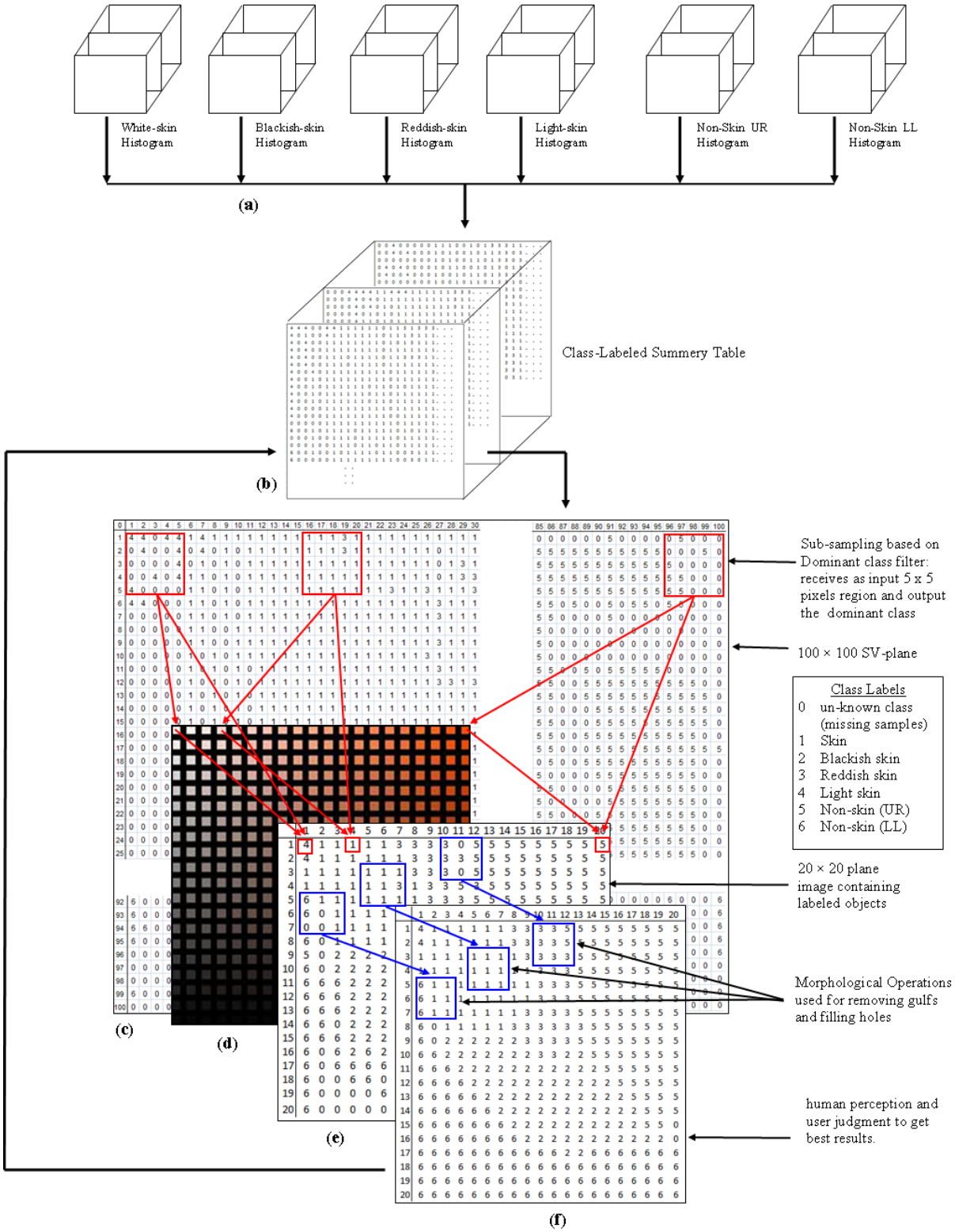


Figure 4.40: The proposed algorithm; (a) constructing a 3D-histogram for each class; (b) class-labeled summary table (c) projected 2D-space of class-labeled summary table; (d) sub-sampling SV-plane of HSV color space; (e) sub-sampling the class labeled SV-plane using dominant-class filter; (f) morphological operations used to remove noise, filling thin gulfs, filling holes, and smoothing borders.

To evaluate the feasibility of the classification boundaries of our approach, the proposed standard set of test images is used and the results are shown in Figure 4.41. As shown in this figure, each test image is shown along with two graphs; these are: the classification boundaries as two-class classification problem (i.e. skin and non-skin), and the classification boundaries using multi-skin modeling (i.e. 4-skin classes graph). The visual inspection shows that these boundaries are feasible and fulfill the general guidelines as described in Section 4.10.2. They are simple, smooth, and with no gulfs. In addition, the boundaries match the actual distribution of the training data and yield compact regions.

The pixel-based quantitative results of the proposed approach are shown in Table 4.10. This table shows the classification results for each hue slide separately using our raw data. Table 4.11 shows the general system performance. The system is capable to detect skin regions and achieve detection accuracy of 98.51878% with a recall of 99.3636% at significantly low FNR of 0.6363% and FPR of 2.04006%. These quantitative results have shown the effectiveness and robustness of the proposed method. The detailed comparison of the proposed method with other skin detection methods is presented in the next section.

For qualitative evaluation, the skin detection results using FEI and CVL face databases are illustrated in Figure 4.42 and Figure 4.43 respectively. Figure 4.44 shows skin detection results using LFW and FSKTM face database.

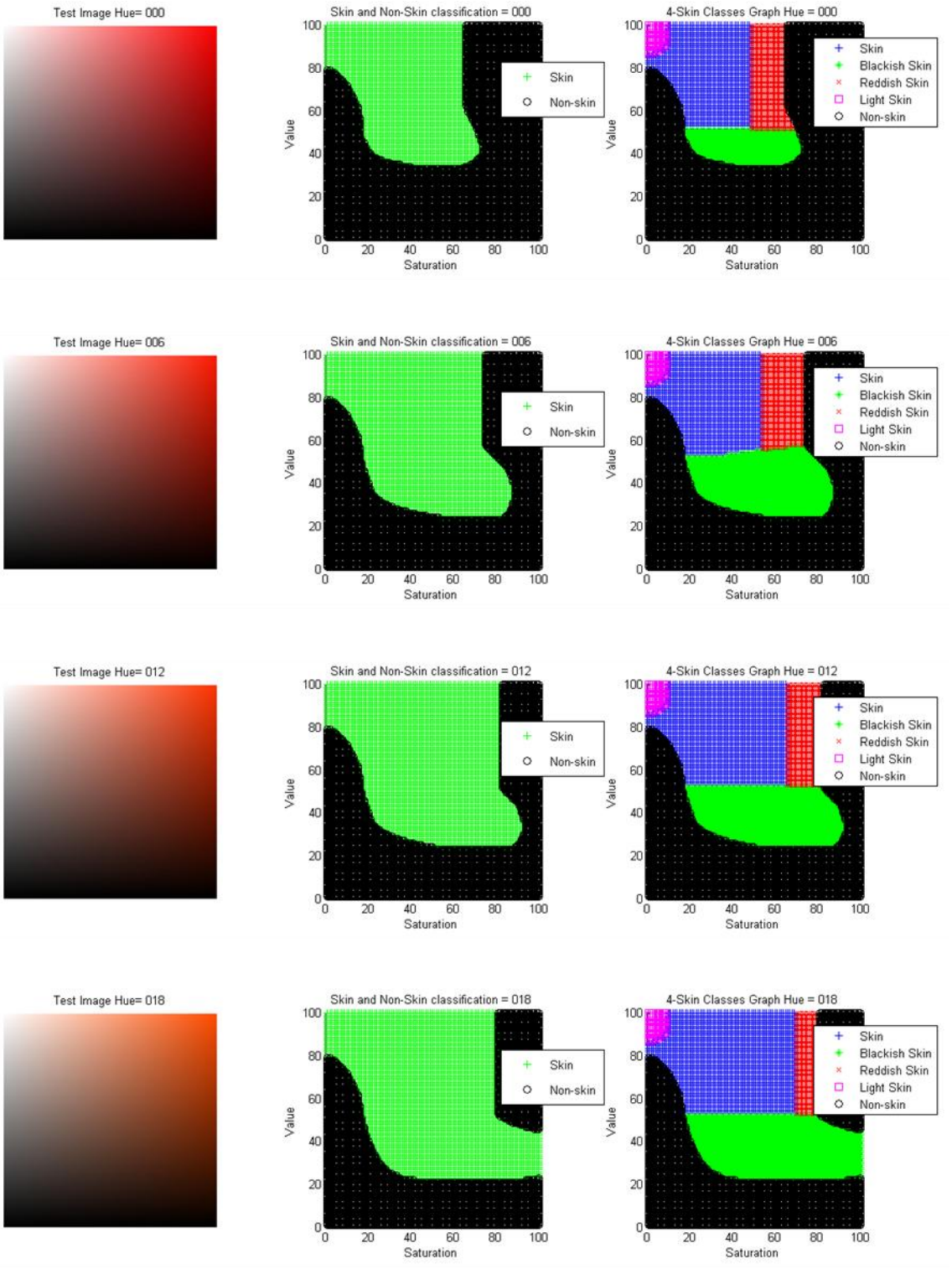


Figure 4.41: Skin detection results of the proposed algorithm;
(a) input image; (b) skin detection as two-class classification problem; (c) skin-detection using multi-skin modeling.

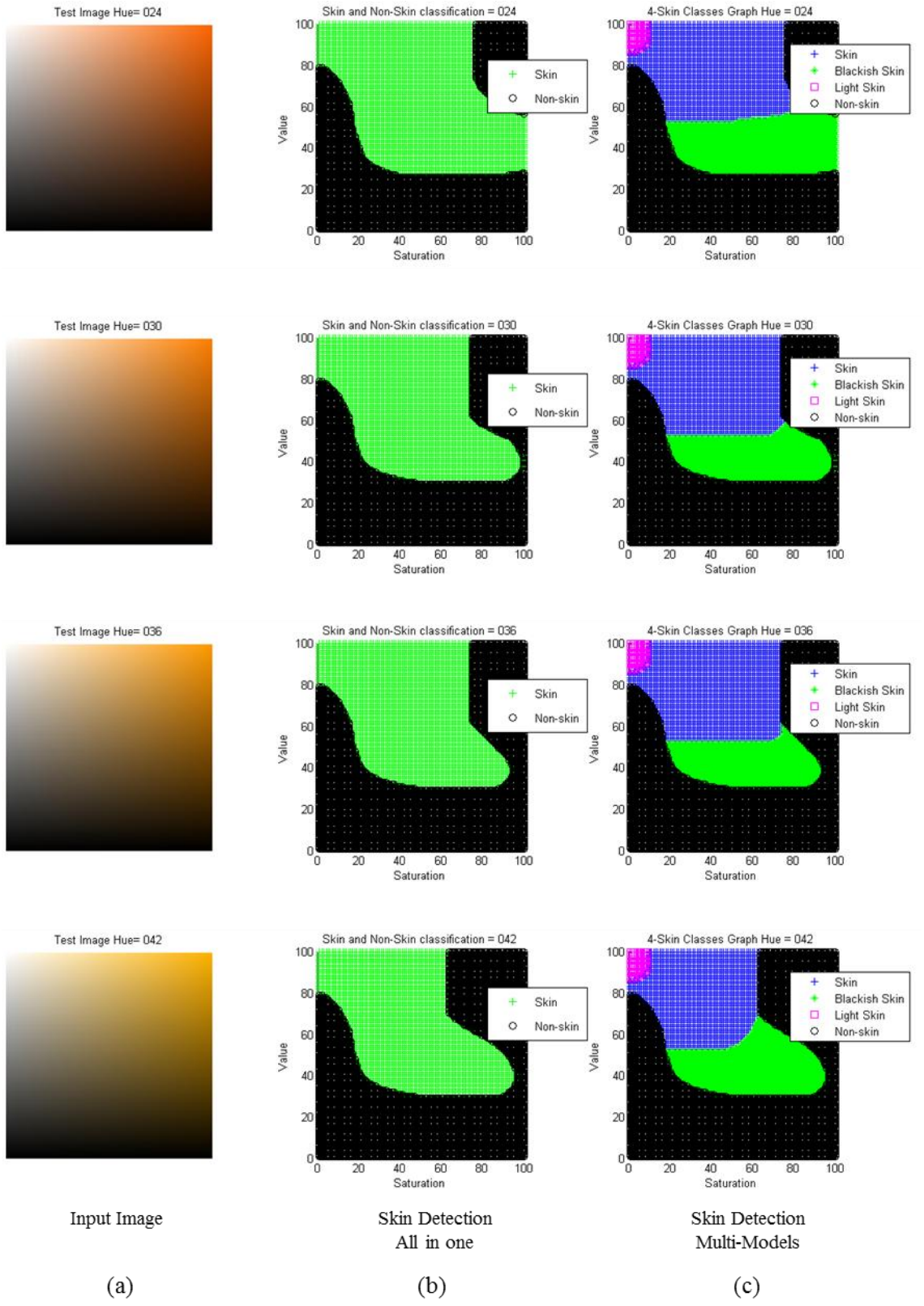


Figure 4.41(*Continued)*: Skin detection results of the proposed algorithm; (a) input image; (b) skin detection as two-class classification problem; (c) skin-detection using multi-skin modeling.

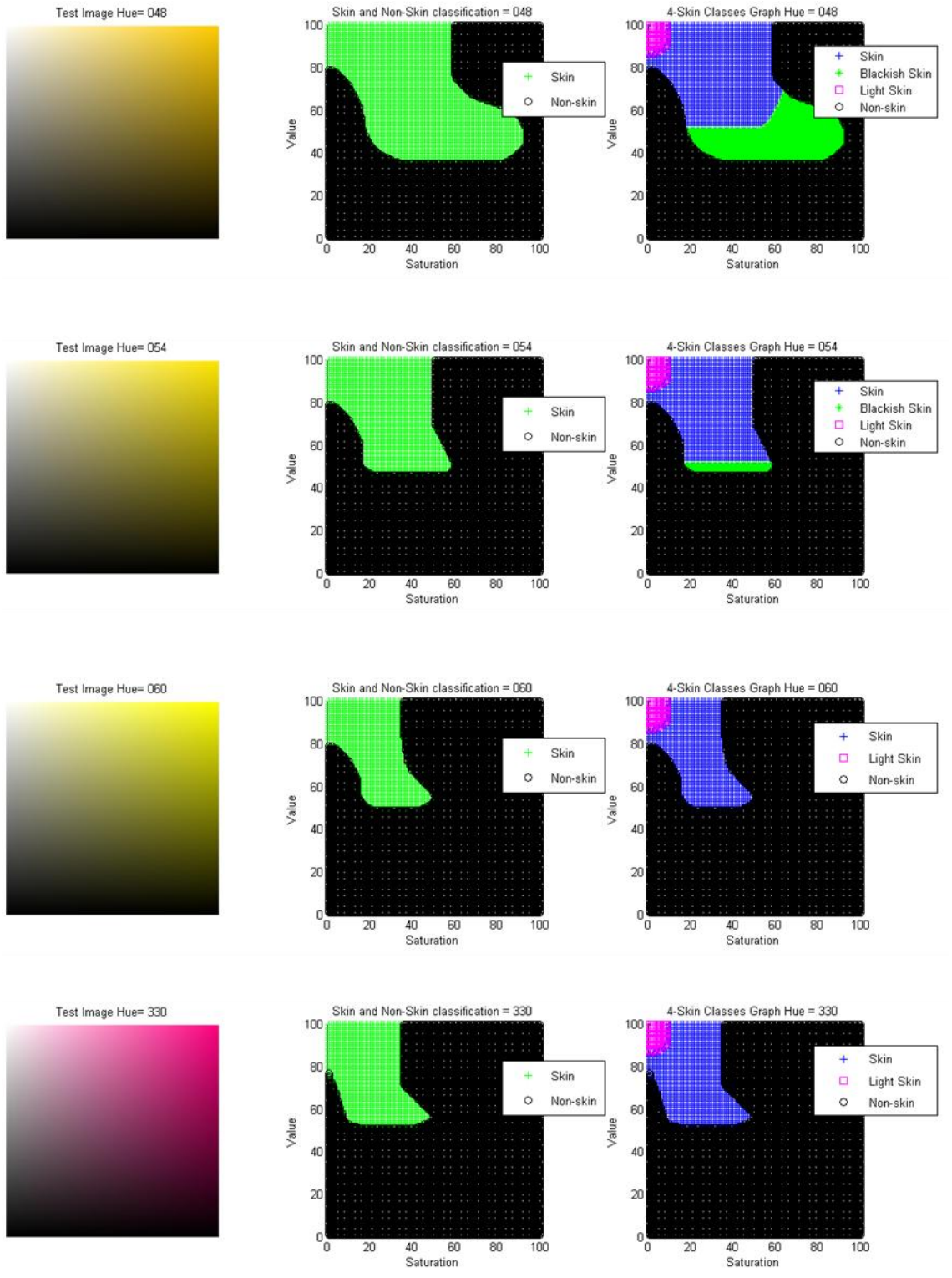


Figure 4.41(*Continued)*: Skin detection results of the proposed algorithm; (a) input image; (b) skin detection as two-class classification problem; (c) skin-detection using multi-skin modeling.

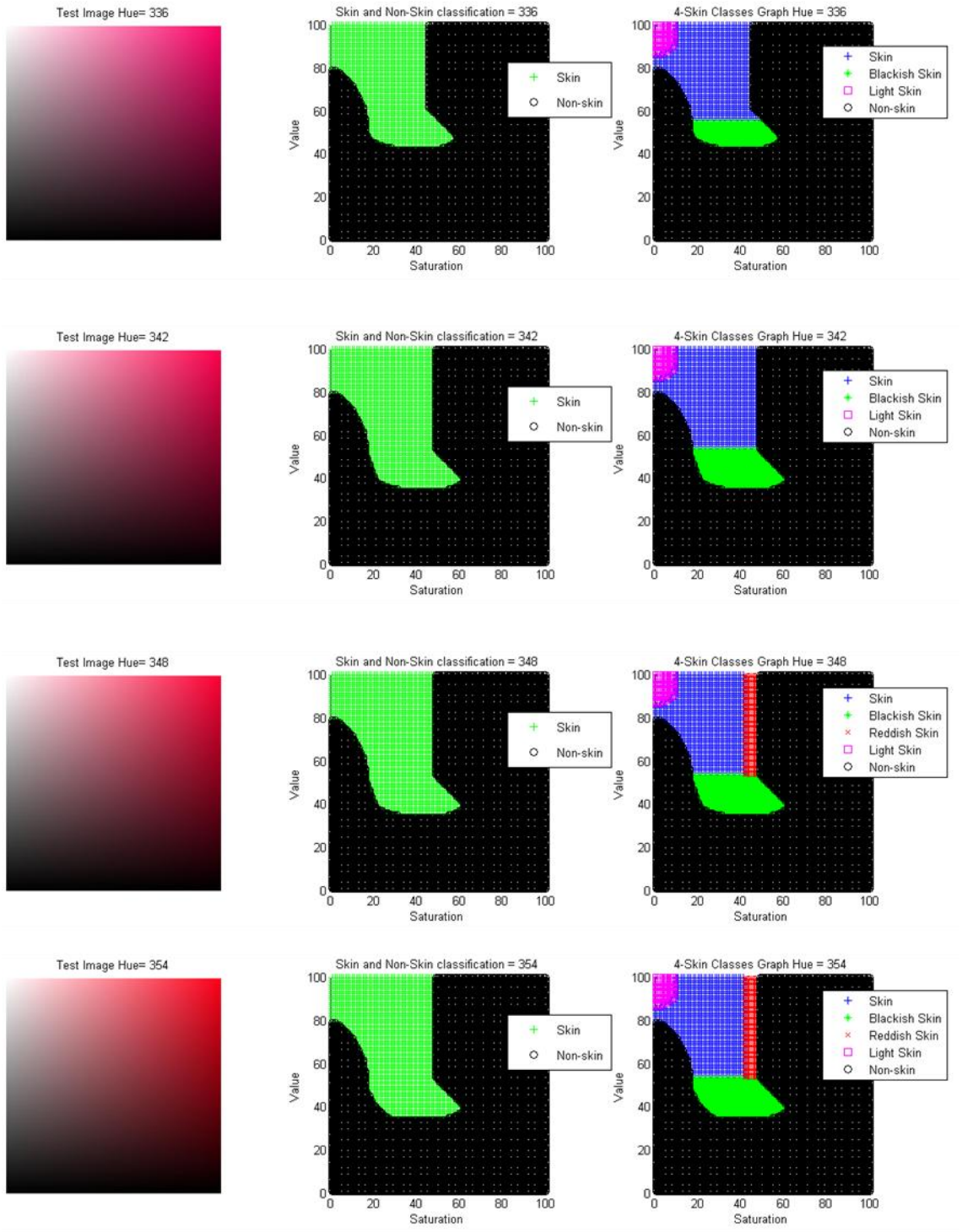


Figure 4.41*(Continued)*: Skin detection results of the proposed algorithm; (a) input image; (b) skin detection as two-class classification problem; (c) skin-detection using multi-skin modeling.

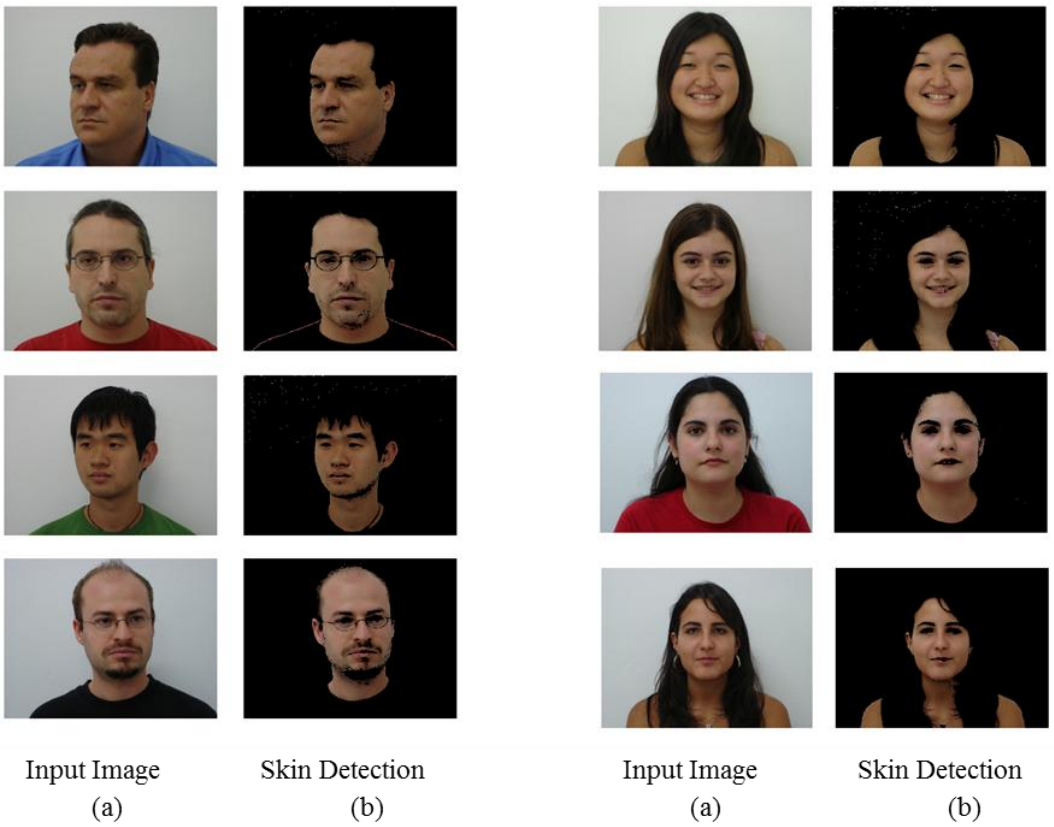


Figure 4.42: Skin detection using FEI face database; (a) input image; (b) skin detection result.

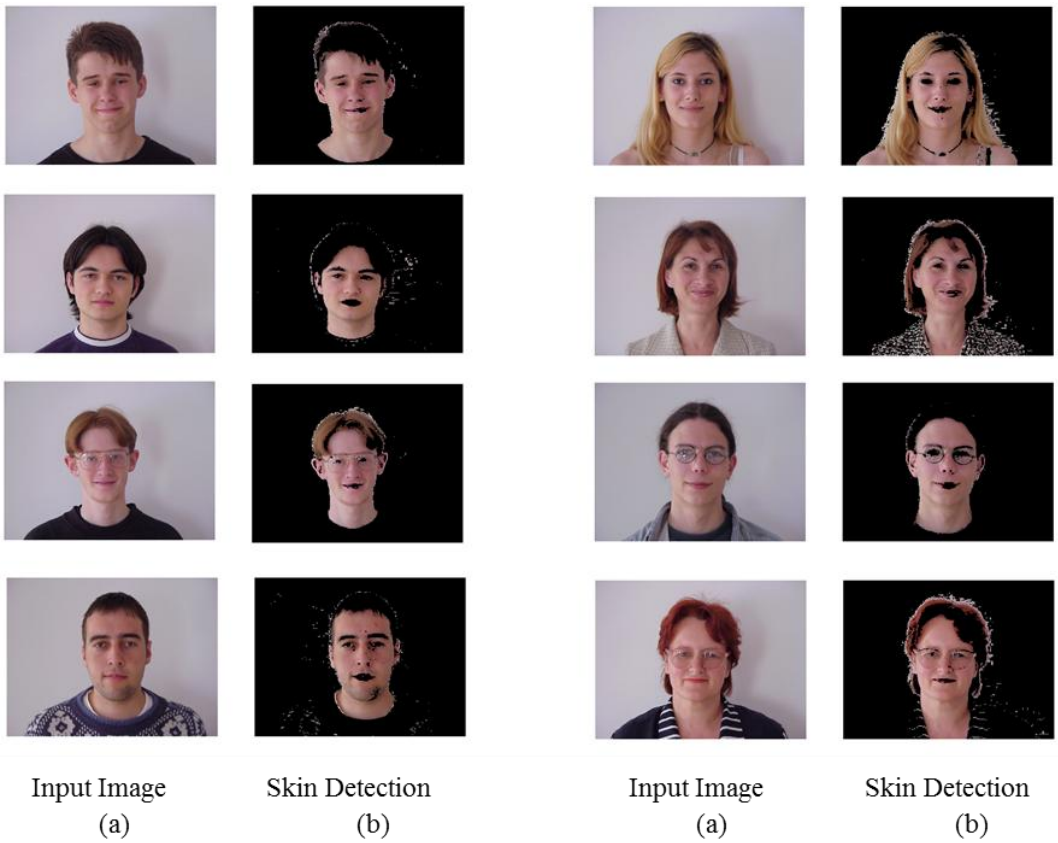


Figure 4.43: Skin detection using CVL face database; (a) input image; (b) skin detection result.

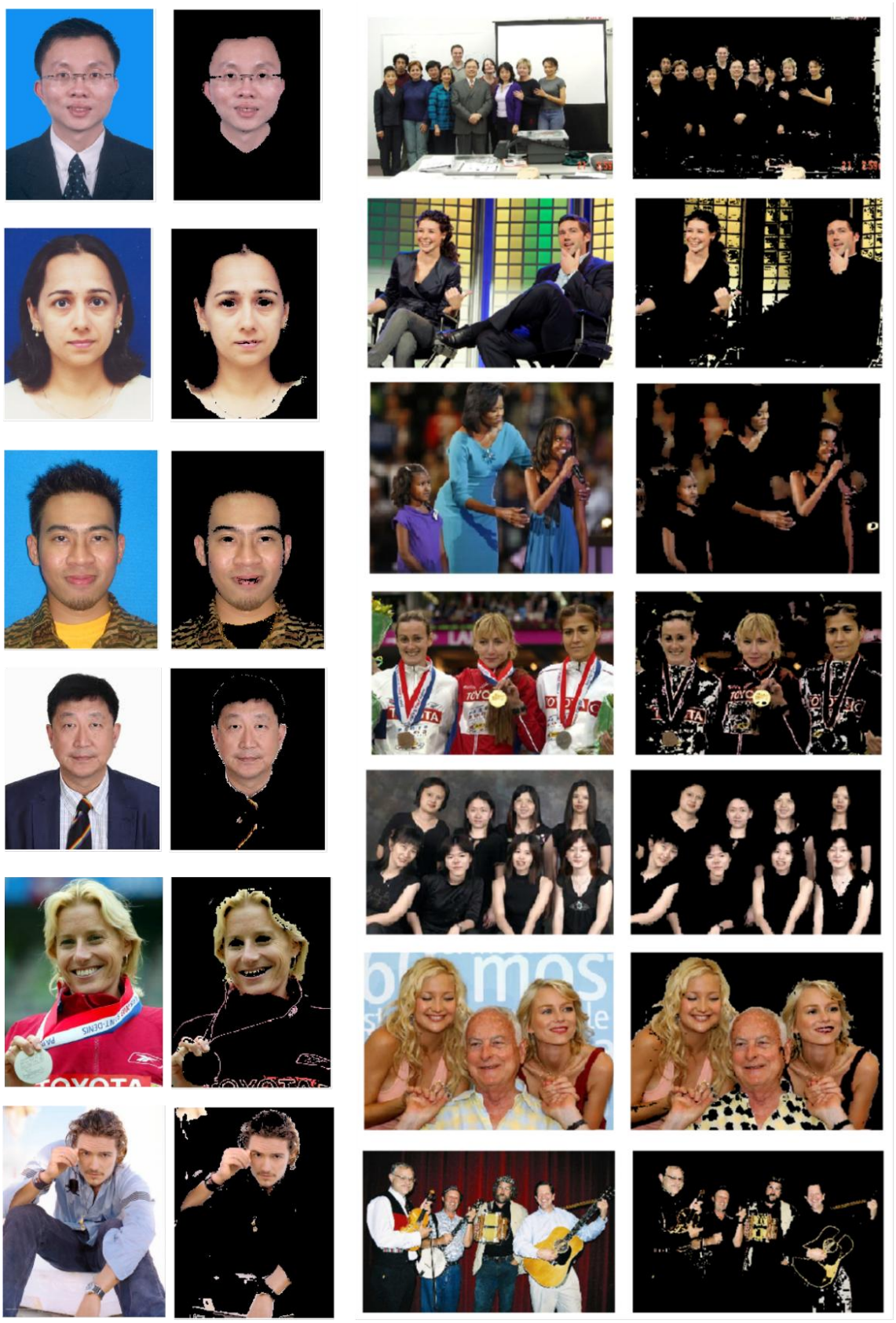


Figure 4.44: Skin detection using LFW and FSKTM databases;
(a) input image; (b) skin detection results.

Table 4.10: Pixel-based quantitative results of the proposed approach using raw data and based on multi-skin color models.

Seq.	Hue	No. of Pixels	TP	TN	FP	FN	FNR %	FPR %	ACCURACY %	RECALL %
1	0	1,996,298	332,086	1,631,924	14,308	17,980	5.1362	0.8691	98.3826	94.8638
2	6	741,311	486,880	212,628	40,355	1,448	0.2965	15.9517	94.3609	99.7035
3	12	1,723,777	994,726	572,910	150,941	5,200	0.5200	20.8525	90.9419	99.4800
4	18	2,999,401	2,738,213	218,866	36,427	5,895	0.2148	14.2687	98.5890	99.7852
5	24	3,185,737	2,686,703	486,102	5,600	7,332	0.2722	1.1389	99.5941	99.7278
6	30	1,402,525	1,134,600	263,181	1,758	2,986	0.2625	0.6635	99.6618	99.7375
7	36	588,969	225,278	349,769	12,674	1,248	0.5509	3.4968	97.6362	99.4491
8	42	1,030,192	122,447	900,386	5,960	1,399	1.1296	0.6576	99.2857	98.8704
9	48	770,064	113,660	655,856	454	94	0.0826	0.0692	99.9288	99.9174
10	54	565,438	31,653	532,377	1,344	64	0.2018	0.2518	99.7510	99.7982
11	60	859,448	40,402	818,747	59	240	0.5905	0.0072	99.9652	99.4095
12	330	684,279	38,163	646,112	2	2	0.0052	0.0003	99.9994	99.9948
13	336	2,184,265	12,754	2,166,390	5,049	72	0.5614	0.2325	99.7656	99.4386
14	342	2,186,998	22,155	2,162,416	1,697	730	3.1899	0.0784	99.8890	96.8101
15	348	1,525,651	71,839	1,446,108	7,063	641	0.8844	0.4860	99.4950	99.1156
16	354	777,651	134,498	628,206	1,451	13,496	9.1193	0.2304	98.0779	90.8807

Table 4.11: The general performance of the proposed approach using raw data

Seq.	Hue	No. of Pixels	TP	TN	FP	FN	FNR %	FPR %	ACCURACY %	RECALL %
1	ALL	23,222,004	9,186,057	13,691,978	285,142	58,827	0.63632	2.040063	98.51878	99.36368

4.12 Comparison with Other Works

For qualitative comparison with other works, Figure 4.45 shows the experimental results of the proposed method applied on real images along with different state-of-the-art skin detection methods; columns from left to right: source images, Solina's method, Chen's method, Baskan's method, Garcia's method, and our method. The images in this figure contain different skin color tones with many variations. The first row shows lighted skin. Rows 2 to 4 show skin colors that tend to blue, pink, and green. Rows 5 to 8 show skin colors with low lighting conditions and/or shadows. The last two rows show yellow and reddish skin tones. However, it can be qualitatively noticed that the proposed method overcomes the sensitivity to these variations which makes it more reliable to detect human skin in complex images.

Figure 4.45: Comparison among different skin detection methods applied on real images; (a) input images; (b) Solina's method; (c) Chen's method; (d) Baskan's method; (e) Garcia's method; (f) our method.

A quantitative comparison of our method with other works is shown in Table 4.12. This table gives an overall view of the performance of different methods.

Figure 4.46 shows the column chart of the general performance of different skin detection methods. By considering the results presented this figure, one can notice that although we used the same method (i.e. Bayes classifier) and the same training data, the detection accuracy of Bayes classifier based on uni-skin modeling is 93.882% with a recall of 86.776% while it is 97.267% with a recall of 95.609% based on multi-skin modeling. This means that the proposed multi-skin modeling improves the prediction accuracy of the classifier.

As shown in Figure 4.46, the proposed method is superior in terms of accuracy and recall rates. Furthermore, the proposed method significantly has the lower FNR in comparison to other methods. Thus, the quantitative results show the effectiveness and robustness of the proposed method to detect human skin in images.

Table 4.12: Performance of our skin detection method compared to other methods.

Method	TP	TN	FP	FN	FNR %	FPR %	ACCURACY %	RECALL %
Solina	9,147,005	11,703,299	2,833,539	644,827	6.585	19.492	85.703	93.415
Chen	6,280,440	14,536,838	-	3,511,392	35.860	0.000	85.567	64.140
Bayes 2-class	8,496,916	14,343,413	193,425	1,294,916	13.224	1.331	93.882	86.776
Bayes Multi-class	9,361,881	14,302,008	234,830	429,951	4.391	1.615	97.267	95.609
Garcia	6,956,050	14,466,457	70,381	2,835,782	28.961	0.484	88.055	71.039
LDA	8,034,671	9,092,880	5,180,000	1,721,444	17.645	36.293	71.279	82.355
Proposed Method	9,186,057	13,691,978	285,142	58,827	0.636	2.040	98.519	99.364

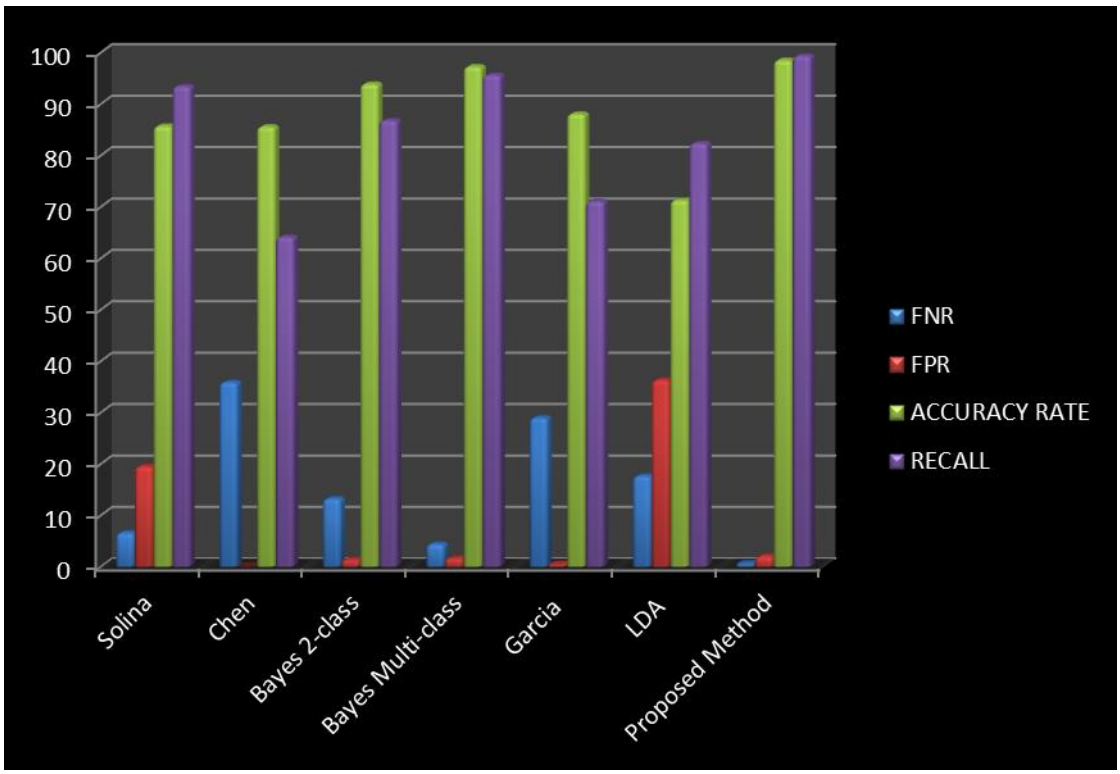


Figure 4.46: Performance of different skin detection methods.

A detailed comparison with four other methods is presented hereby.

i) Chen & Wang (2007) proposed a two-staged region-based and content adaptive skin detector. In the first stage the detector applies complete region-based image segmentation based on color-texture to segment the source image into homogeneous regions. Then in the second stage, a pixel-based segmentation is applied to extract the candidates "key skin region" through a set of rules in the RGB color space.

The main differences can be summarized as follows:

- In our work we carry out pixel-based segmentation followed by region-based segmentation; whereas Chen's approach starts with complete region-based segmentation followed by pixel-based segmentation. This makes our current approach faster and more reliable because partial segmentation is faster than complete image segmentation. The difference between complete and partial image segmentation was discussed in detail in Section 3.1. Figure 4.47 shows an example adopted from Chen's work (2007). The figure shows a lot of needless work was done to segment the

background. However, there is no need for complete image segmentation because it implies that every image pixel must be in a region. This means that complete segmentation algorithm should not terminate until every pixel is processed and each pixel may be processed many times (i.e. including background). This implies high computational cost, which makes it impractical in many applications. Conversely, partial image segmentation usually stops when the regions of interest are isolated. This would reduce the size of data to be processed because most of the colors in color space are non-skin colors. The substantial reduction in data volume offers an immediate gain in speed, which is important for most applications. Furthermore, complete image segmentation implies many sub-problems. For instance, suppose that a single region of an object is segmented into more than one sub-region. Gathering many sub-regions to represent that object may be failed.

- The second difference as stated in Chen's work is that about the usage of two color spaces, the RGB and YCbCr. The former color space is first used to detect skin regions. Then the YCbCr color space is used to overcome the limitation of RGB. It is known that RGB is suitable for technical applications, but it is of limited use for image segmentation and analysis because of the high correlation between the R, G, and B components, and is not a perceptual model (Efford, 2000).

  From our point of view, there is no need to use two color spaces. In contrast, our approach is based on only HSV color space, which is already a perceptual model.

- Their system used uni-skin model, which is unable to detect variations in skin tones. In contrast, a key contribution of this work is by shifting to multi-skin color models. Therefore for images with multi-face of different ethnic groups and illumination variations, the algorithm described in this research is expected to outperform that of Chen's approach. The comparison in Table 4.12 shows that in addition to higher detection rate, our system is also more robust to variations in skin color (i.e. lower false negative rate).

- Testing and evaluation procedure shows experimentally that the classification boundaries of Chen's approach are infeasible and show very low detection rate as in Figure 4.25 and Figure 4.26.

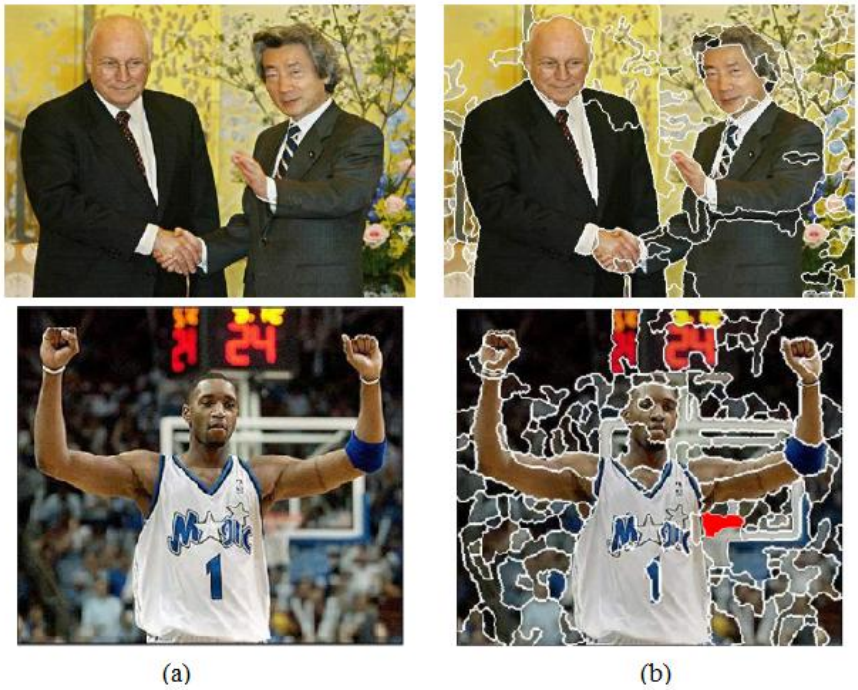


Figure 4.47: Complete image segmentation approach proposed by Chen (2007); (a) input image; (b) region-based image segmentation implies high computational cost as a lot of needless work was done to segment the background.

ii) The Cho's segmentation approach (Cho *et al.*, 2001), is capable of adaptively adjusting its threshold values and electively separating skin color regions from background. The system first chooses rough upper and lower threshold values for each color component (hue, saturation, and value) by observing the skin color distributions of several sample images. In the HSV color space these threshold values define a 3D box, called the *thresholding box*. The color vectors inside the thresholding box play the role of standard color vectors representing normal skin color. Then, a new thresholding box is formed with the updated threshold values. The following are the limitations and drawbacks of Cho's segmentation approach:

- Cho's approach (2001) assumes that the color of the face is dominant. By dominant color, they mean that the region of that color, i.e. face, is larger than those of other colors. The authors stated: " if an image is given whose background color is much more

dominant than the color of the true skin regions and the difference between the two colors is relatively large, the method will fail" (Cho *et al.*, 2001).

- Cho's segmentation approach will not be able to find the skin color regions if an input image is taken for a group of people that are composed of several different races (Cho *et al.*, 2001). However, it is theoretically possible to detect them if the method is applied sequentially with different initial settings.
- Cho's approach is based on *thresholding box* in HSV color space. However, thresholding box has the following limitation: if the actual distribution of skin color values has some other shapes in the color space, for instance if it is stretched out in a direction that is not parallel to one axis, then this box is inadequate to select the desired range of the real color distribution.

  However, our method does not have the above-mentioned limitations. For example:

  The method can detect skin regions independent of size, shape, location, number of faces, and complex background. By using multi-skin models, the method overcomes the sensitivity to variations in lighting conditions and ethnicity. Images that are taken for a group of people that is composed of several different races, is segmented in different layers. There is no need to apply the method sequentially with different initial setting because we already have multi skin models. The clusters in our method do not have a fixed shape. The models vary in the color space and can be stretched out or shrunk in any direction as shown in Section 4.11.

iii) Chen H. *et al.* (2008) have used Bayesian approach to determine the face-color region in RGB. This approach assumes that the color distribution of the human face is different from that of the image background.

The following are the limitations of this method:

- The results are not reliable under the conditions of complex background or uneven illumination (H. Y. Chen *et al.*, 2008). It is known that RGB color space is of limited

use for image segmentation and analysis because of the high correlation between the R, G, and B components (not perceptual model).

- The probability theory is computationally expensive. For each input pixel, a probability value is calculated to indicate its likelihood of belonging to the skin color. Then skin likelihood probabilities for the whole image are normalized into a specific range. The next step is to segment the likelihood image into skin and non-skin regions using thresholding (Shih, 2010) whereas, the approach in this research uses Look-up table which is faster because all the required calculations are done offline.
- Chen H. *et al.* (H. Y. Chen *et al.*, 2008) have stated that "the constraints of shape and size of face region are applied on each candidate face area to find the potential human faces". In general, preconditions and constraints make the method of limited use.

iv) Tan et al. (2012) proposed a human skin detection approach that consists of four steps. First, the system detects human faces from the source image. Second, a dynamic method is used to estimate the skin threshold value(s) on the detected face(s) regions using log opponent chromaticity (LO) color space. Then, the 2-D histogram with smoothed densities and Gaussian model are used to define the distribution of skin and non-skin. Finally, a fusion framework that uses the product rule on the two features is employed to obtain better skin detection results.

The main differences of our approach can be summarized by the following:

- The goal of our skin detection system is to solve face detection problem. We divided the principal problem into several manageable sub-problems that, when solved, will resolve the main problem. Skin detection step is used to locate a set of candidate face regions. These regions need further subsequent processing steps such as a complex classifier to make the final arbitration (i.e. face or non-face). Whereas Tan's approach do the opposite. First, the approach detects human faces and then returned back to detect the skin color. It is clear that detecting human faces is a principal problem. Therefore, when no face is detected, this method may fail to detect skin regions. On

the other hand, our approach has the ability to detect all skin regions even when no face is present.

- The computational cost of Tan's approach is very high. According to (Fasel, Fortenberry, & Movellan, 2005), detecting faces in 8000 images requires the system to examine about 1 billion sub-image windows (i.e. about 125,000 sub-image per image). According to (Rowley et al., 1998), detecting faces in 130 images requires the system to examine about 83,000,000 sub-image windows. All these sub-image widows have to be classified. This makes this method impractical for skin detection where the time is a critical factor. In contrast, our method is very fast because it is based on Lookup-Table to classify image's pixels (i.e. skin or non-skin).
- Tan's approach estimates the skin distribution and thresholds based on the detected faces. When the input image contains people from different ethnic groups, the system cannot adopt itself to change the thresholds for the classification. In contrast, our system already shifted to multi-skin modeling to cover different ethnic groups and illumination.
- Tan's approach uses 2D histogram to define distribution of skin color due to ignoring intensity component. In general, discarding any color information affects the model's accuracy. Our approach uses the full color information that makes it more accurate to detect skin pixels.

4.13 Applicability of the Proposed Approach for Other Applications

The proposed approach shows the potential to be applied to a range of applications such as face recognition, video surveillance, naked images filtering, teleconferencing, and hand gesture recognition. Figure 4.48 show three main applications of our system applied on real images. For instance, filtering adult-content images is very important for search engines to avoid offensive content on the Web. Skin detectors can provide an efficient and effective way to detect naked people in images (Lee, Kuo et al. 2007) (Duan, Cui et al. 2002).

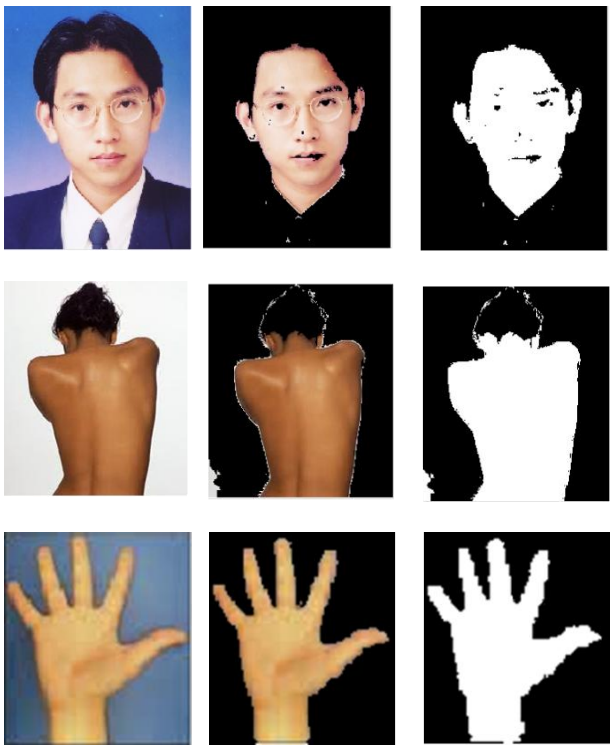


Figure 4.48: Three main applications of skin-color detection.

4.14 Summary

Using color information as a cue feature for face detection denotes three main pertaining subproblems due to variations in illumination, ethnic groups, and camera characteristics. Thus, in order to employ a color-based approach for face detection, we have to find suitable solutions to mitigate these problems. Most of the published approaches suffer either from their high computational cost or seem to lack robustness against the above-mentioned variations.

In this chapter, we presented a reliable skin-color detection approach for automatic locating skin regions in image(s) without calculations which makes it very fast. Our approach is based on using SD-LUT lookup table. The quantitative and qualitative results demonstrated the effectiveness of the proposed approach to produce a consistent detection rate despite variations in many random factors such as illumination, different ethnic groups, and camera characteristics.

Based on the contents of the SD-LUT entries, we found that the number of skin color entries form about 9.64% of the SD-LUT. In other words, most of colors in the color space are non-skin colors which form about 90.36%.

Our approach is based on building multi-skin color clustering models using the HSV color space. Four skin models are used, namely, the white-skin, blackish-skin, light-skin, and reddish-skin models. In general, if the objective is to locate faces of a particular race, one should create skin color model that use skin samples from only that race. With such models, the system improves the accuracy of skin detection despite wide variation in ethnicity and illumination. To the best of our knowledge, this is the first attempt that employs multi-skin model for skin detection. The proposed approach reduces computational cost (i.e. runs without arithmetic operations) as the classification boundaries are transformed into Lookup-Table which is constructed offline and indexed by a color information vector. Since color information is robust against scale, position, and orientation, the method can detect human faces of different sizes and orientations in image(s).

Due to the limitations about how to measure segmentation accuracy and error rates, in this chapter we have presented a novel method for testing and evaluating the quality of image segmentation algorithms based on color feature and provided step-by-step framework in which to use and compare between different methods. A set of standard test images were proposed to get fair and accurate evaluation. Additionally, we have provided detailed examples of such evaluation.

From our point of view, previous works built the skin color model as a separate stand-alone unit in order to get the best skin segmentation performance. For example, many face detection systems deal with skin detection and illumination correction as separate goals. In this work, we look at this problem in a different way based on the simple fact that building skin models is not the final goal but merely a part of a complete integrated system. So, rather than treating skin detection and illumination correction as separate goals, we have attempted to develop a unified approach. Recognizing this, building skin color models should take into consideration the other stages of the system, and consequently the way to build skin color models might change accordingly. As will be shown in the next chapter, illumination correction can be done in a simple, fast, and direct way.

CHAPTER FIVE

ILLUMINATION ENHANCEMENT METHODOLOGY

5.1 Introduction

Illumination variation is highly important and difficult problem in face detection. In general, human face and other parts of human body recorded under unconstrained imaging conditions are frequently subject to illumination variations which affect their appearance. It has been shown that varying illumination and shadows cast by facial features may cause many subproblems such as false color tone, loss of feature information, shape distortion of objects and failure of conjugate image matching within the shadow area.

As mentioned in Section 2.6, problems caused by illumination in color imaging are handled in general in three different manners (Martinkauppi 2002): 1) preventing changes by controlling illumination or ignoring information taken under changed condition; 2) using a process which disregards the effects of illumination variations such as *image enhancement pre-processing* and/or *color consistency* techniques, 3) using *features* and/or *face models* that are relatively insensitive to illumination changes. The first option is inadequate in many applications because it is impossible to control illumination in many real world situations. On the other side, ignoring information may lead to a loss of essential data. In many real-time applications, the developers usually control the environment in which the detection system will take place. Many lighting defects can be minimized by careful setup of imaging conditions, or if they cannot be eliminated altogether, the defects can be assumed to be constant over some period of time.

With the second option, although many approaches and theories have been proposed for illumination correction and color constancy, none has been proven to produce generally successful results for all kind of images (J. B. Martinkauppi & Pietikäinen, 2005). In fact, it has been shown by Agarwal *et al.* (2006) that most machine color constancy approaches cannot

handle situations with more than one illumination present in the scene. Furthermore, these techniques imply high computation cost.

With the third option, An *et al.* (2010) showed that the illumination invariant features are limited and not enough for recognition in large scale face databases. Under the largely varied lighting conditions, these methods achieve low level performance.

With the diversity of complex images types and source, dealing with illumination variations in application to face detection becomes harder. From our point of view, the correct procedure is the *image enhancement pre-processing* techniques. In general, the best transformations for image enhancement normally are selected interactively. The idea is to adjust experimentally the image brightness and contrast to provide maximum detail over a suitable range of intensities. Unlike the interactive enhancement, the transformations can be applied to color images in an automated way. As mentioned before, automatic image enhancement can be global or local. Each one has its drawbacks. Global methods increase the visibility of one portion, aspect, or component of an image, generally by suppressing others, whose visibility is diminished. On the other hand, local image enhancement in application to face detection problem causes serious delay to the operation of the appearance-based classifier.

In this chapter, a novel method for automatic illumination enhancement is proposed. The method transforms the face image captured under non-uniform lighting conditions into a new face image as if it has been captured under near-uniform lighting conditions. The method is fast, simple, reliable, and free of tuning parameters. The new generated lighting-corrected image will not replace the source image but it will be used along with the source image in order to improve the performance of the appearance-based classifier (Chapter 7).

5.2 Methodology of Skin Color Enhancement

In this work, we take into account that skin segmentation and skin color correction are so closely related that they should not be performed separately, and as somewhat related to the work by Hsu *et al.* (2002). Therefore, it is proposed that illumination correction process is done *locally* to "regions of interest" along with image segmentation step. Dealing with these two steps as different separated goals makes the problem harder.

In practice, it has been found that local illumination correction can drastically improve the visibility of some facial features while leaving the other regions in the image unchanged. In this research, the goal is to adjust the darkness level of some pixels (or face region) to provide maximum detail over a suitable range of intensities. The algorithm is of three steps:

- Estimate the illuminant of the facial region from the skin segmentation results.
- Based on this estimation, enhance the illumination of the other local facial regions which are dark.
- Generate a new face image as if it has been captured under near-uniform lighting conditions.

As shown in Chapter 4, our skin segmentation approach is based on using multi-skin models, namely: white-skin, shadow-skin, light-skin, and reddish-skin models. An important advantage of multi-skin modeling is to exploit more information about different skin tones and the relationship between them. As the segmentation output is kept in multi-layer binary images (i.e. L_1, L_2, L_3, and L_4), it is easy to carry out automatic color enhancement using HSV color space. For instance, let us consider an example of non-uniform lighting shown in Figure 5.1. As is evident in the example, the left side of the face is illuminated with normal lighting while the right side of the face is illuminated with low lighting.

Figure 5.1: Example of non-uniform lighting.

In this research, the idea is to adjust the illumination of the right side of the human face to match the illumination of the left side. Therefore, the approach does not need to be performed on the entire image. In other words, the skin regions identified at Layer L_2 (shadow-skin) should appear similar to the skin regions identified at Layer L_1(white-skin). Therefore, we have to lightening the right side of the face based on estimating the illumination of the left side.

Recognizing that, the correct procedure is to work on HSV color space, leaving the color information unchanged and processing just the brightness or luminance values.

Given that the brightness of the color in the HSV color space is represented separately in the V component, we can adjust the brightness of a pixel in different ways such as adding a constant bias b to each pixel brightness (Efford, 2000):

$$G1(x, y) = V(x, y) + b \tag{5.1}$$

If $b > 0$, the overall brightness is increased. Similarly, we can adjust the contrast through multiplication of pixels brightness by a constant gain a:

$$G2(x, y) = a * V(x, y) \tag{5.2}$$

Equations 5.1 and 5.2 can be combined to get a general equation for adjusting both brightness and contrast of an image as follows:

$$G3(x, y) = a * V(x, y) + b \tag{5.3}$$

Since we do not want to specify a constant gain and bias, but would rather map a particular range of brightness [f_1,f_2] of skin segments at layer L_2 onto a new range [g_1,g_2] as of skin segments at layer L_1, this form of mapping is done using the following equation , adopted from (Efford, 2000):

$$G4(x,y) = g_1 + \left(\frac{g_2 - g_1}{f_2 - f_1} \right) [V(x,y) - f_1] \quad (5.4)$$

where each of f_1 *and* g_1 is the average intensity of a region minus its standard deviation, each of *f2 and* g_2 is the average intensity of a region plus its standard deviation. In this research, we tested the four aforementioned equations and found that Eq. 5.4 is more general than others because it can be adopted for various types of image.

Figure 5.2 shows the idea of our proposed local illumination enhancement approach. Figure 5.2(a-c) illustrates pixel-based image segmentation using skin color feature. Figure 5.2(d) shows the skin color correction of skin segment at layer L_2 by adjusting the darkness of each pixel in this layer. Then, the iterative merge is used to create a candidate face region is shown in Figure 5.2(e). The newly-generated image (i.e. lighting-corrected image) is shown in Figure 5.2(f). In this figure, it is clear that the right side of the face is lighter than before which can produce substantial improvements in the visibility of details. The original image and the newly-generated one (i.e. lighting-corrected image) would be passed to the subsequent ANN-based face detector (see Chapter 7). In this work, we also found that the saturation channel *S* of HSV color space had a good effect on the skin color enhancement for reddish-skin regions that appear with highly concentrated red color. Thus, we used the same technique to adjust the saturation channel.

Illumination correction is not an easy task in RGB color space due to high correlation between its color components (R, G, and B). Furthermore enhancing RGB colors require to be done on all color components in isolation, but it is relatively simple in HSV as shown.

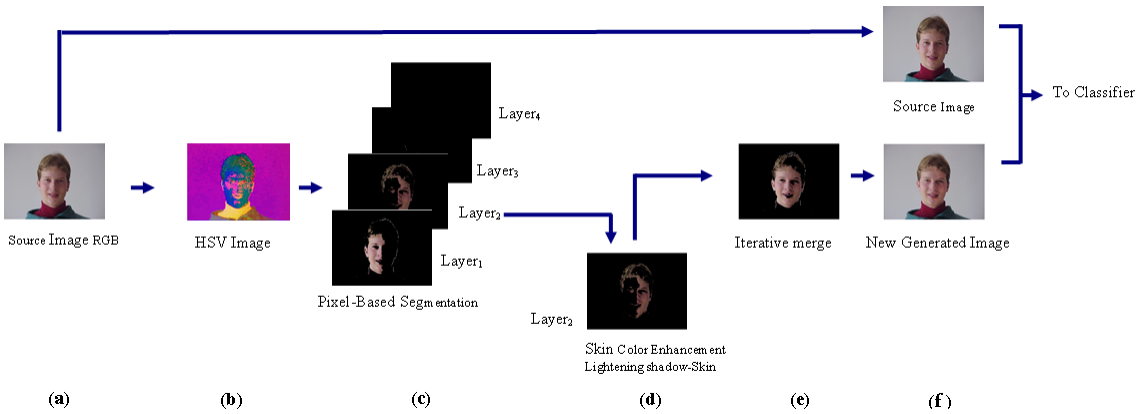


Figure 5.2: Local illumination enhancement; (a) RGB source image; (b) HSV image; (c) pixel-based image segmentation; (d) skin color correction – lightening dark regions; (e) iterative merge; and (f) newly-generated image (i.e. lighting-corrected image).

5.3 Experimental Results

Figure 5.3 shows two examples of illumination correction using CVL dataset with non-uniform lighting conditions. The input color image is shown Figure 5.3(a). The results of pixel-based image segmentation using skin color feature are shown in Figure 5.3(b). The newly-generated image (i.e. lighting-corrected image) is shown in Figure 5.3(c). Figure 5.4 shows four examples of illumination correction using LWF and FSKTM dataset with non-uniform lighting conditions. Although the illumination enhancement may be hardly recognized be humans (visual inspection), the computers deal with intensity levels values (i.e. numbers) which are highly improved in the new lighting-corrected images. Furthermore, the illumination correction is done locally to skin regions rather than the whole image.

The subsequent appearance-based classifier receives two images; the source one and the new lighting-corrected version.

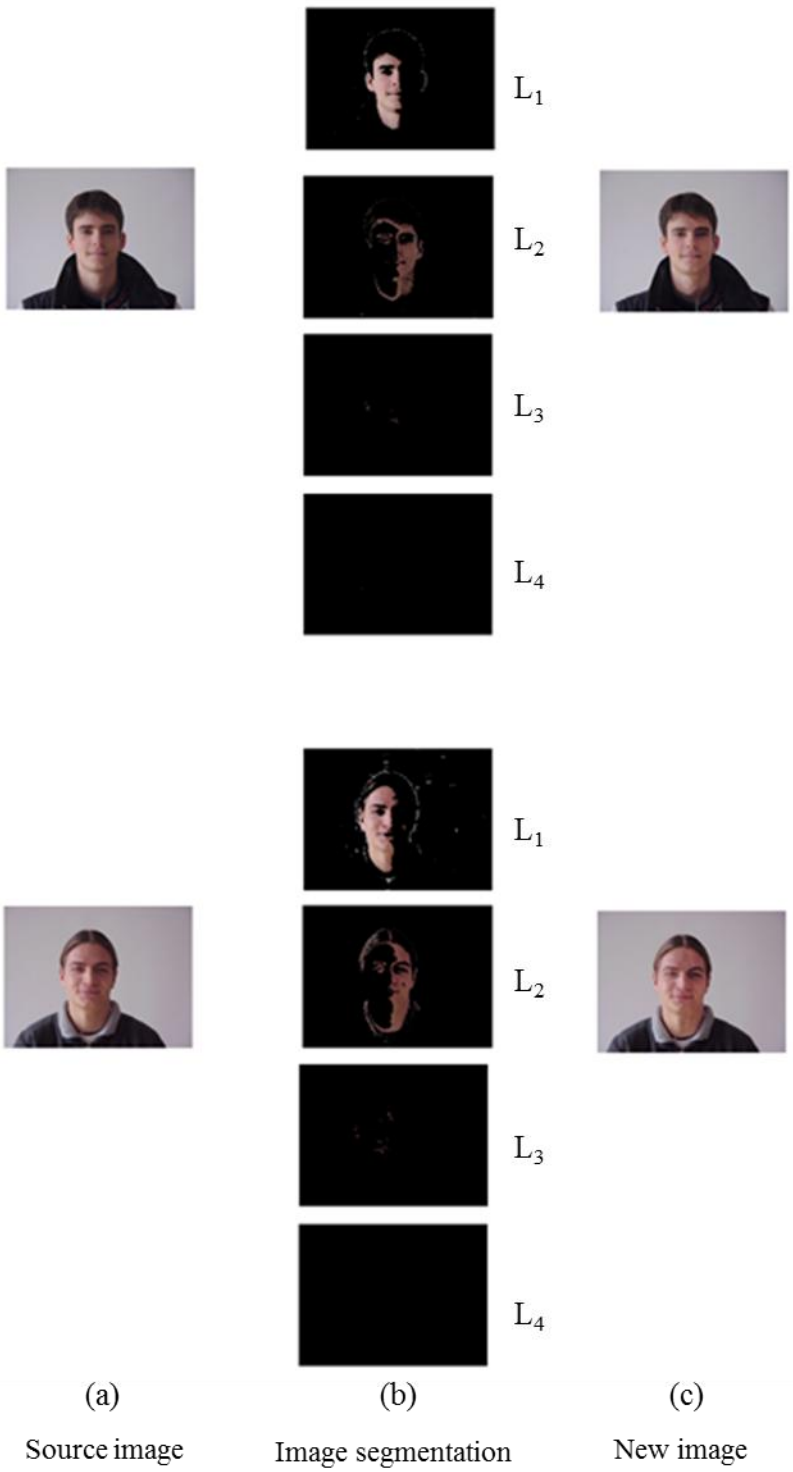


Figure 5.3: Local illumination enhancement using CVL dataset.

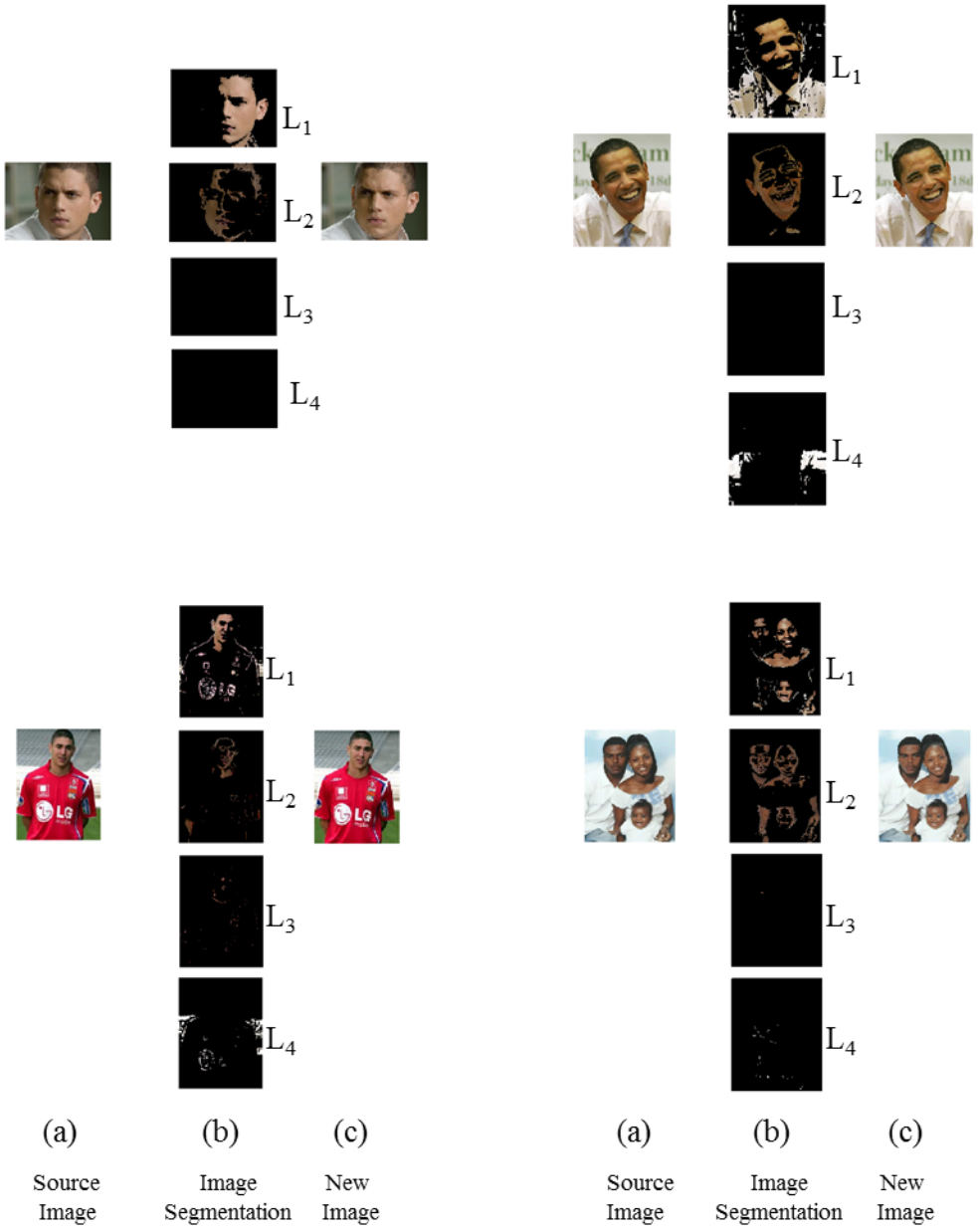


Figure 5.4: Local illumination enhancement using LWF and FSKTM dataset

The main drawback of local enhancement is that it may increase the contrast near edges. Figure 5.5 shows an example of such case. As is evident in the example, the process of local enhancement makes pixels that were dark brighter than their surroundings. This increases the contrast near edges. In this research, to mitigate this sudden change in the intensities at the edges, we propose to smooth the intensity level at the region's boundaries using an average filter of size 3×3 pixels.

.

Figure 5.5: Local illumination may increase contrast near edges; (a) input image; (b) image segmentation; (c) new generated image; (d) abrupt change in intensity level at the borders of regions causes increasing contrast near edges.

5.4 Comparison with Other Works

Although variable illumination is one of the most challenging problems with face detection/recognition (Xie & Lam, 2005) which needs to be dealt with, many previous face detectors ignore this problem or they used some kind of preconditions on the input images.

Rowley *et al.* (1998) proposed a local enhancement approach based on estimating the best fit function. As they used sliding window technique, they proposed to perform lighting correction for each sliding window before passing it to the classifier. As reported in their work, a linear function is used to approximate the overall brightness of each part of the sub-image window, as shown in Figure 5.6, and can be subtracted from the window to compensate for a variety of lighting conditions. Figure 5.6(a) shows examples of sub-image windows; (b) shows best fit linear function; (c) shows the lighting-corrected sub-image.

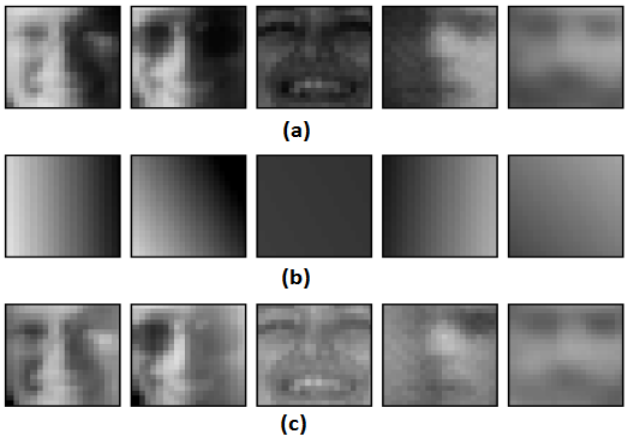


Figure 5.6: Steps of lighting correction approach proposed by Rowley (1998); (a) Original sub-image. (b) best fit linear function. (c) lighting-corrected sub-image

It is clear that this approach implies high computational cost because the lighting correction step is done at the classification phase which implies serious delay to the classifier. In other words, lighting correction procedure would be repeated for each sub-image window (i.e. from about a quarter million to several million depending on image size). In contrast, our system does not have this drawback. In this research work it is proposed to perform the illumination enhancement procedure only once before the classification phase. The idea is to create a new lighting-corrected image in advance as illustrated in Figure 5.2(f). This makes the classifier to run very fast.

Figure 5.7 shows a qualitative comparison between global image enhancement and our local image enhancement approach. As shown in this figure, the results show the effectiveness and robustness of the proposed approach.

Figure 5.7: Global vs. local image enhancement; (a) global image enhancement; (b) local image enhancement; i.e. our approach.

5.5 Discussion

Most of the existing face processing techniques are based on image intensity. However, due to difficulty in controlling the lighting conditions in complex images, variable illumination is one of the most challenging problems with face detection/recognition (Xie & Lam, 2005).

Automatic lighting correction is an important pre-processing step in the computer vision system to improve the performance. Global or local image enhancement may improve the visibility of features (Russ, 2007). Furthermore, it was noticed that when applying image enhancement for images with controlled illumination, the performance of face detection/recognition is substantially improved (Zou et al., 2007a).

In order to perform a reliable color enhancement, suitable color space should be used. In general, it is very difficult to automatically adjust skin color tones in the color spaces such as RGB, CMY(k), etc., because of high correlation between the color; and if it is done using some kind of a transformation function, this means mapping all three (or four) color components with the same transformation function. On the other hand, the HSV color space is an ideal model to deal with color's illumination in images, since the darkness of color is represented separately in the Value *V* component. The colors themselves are not changed.

To cope with the changing illumination conditions, we devised an automatic illumination enhancement technique which is applied along with image segmentation. The approach is fast, simple, reliable, and free of tuning parameters. The approach is based on local enhancement of skin color tone rather than the entire image. The main advantage of this approach is to be done in advance only once before the complex classifier is called.

CHAPTER SIX

FACE-CENTER LOCALIZATION SYSTEM

6.1 Introduction

Human skin detection is an efficient approach for face localization in canonical face images (i.e. used to store face images in databases). With the diversity of image types and sources, skin detection alone is inadequate for automatic human face detection due to two challenges:

1) Images that are captured under unconstrained imaging conditions usually contain objects with skin-like colors.
2) The other exposed parts of the human body such as shoulders, hands, and legs are actual skin regions but they are not faces.

This chapter presents a rule-based geometrical knowledge approach that aims at removing false alarms caused by objects with color similar to skin color. We call this stage "face-center localization system". The ultimate goal is to reduce the search space to more specific "hot spots".

In practice, the existence of facial features such as eyes, nose, and mouth blobs is strong evidence that the potential skin segment is indeed a face (Baskan *et al.*, 2002). Accordingly, the system examines each skin segment looking for the facial features and then estimates the location of the candidate's "face-center". This system module is initialized on a skin-map (the output of the skin detector) and processes it in two steps. The first step is to extract facial features. The second step is to verify the interrelationships between the facial features (i.e. structure) using predefined 2D geometrical rules. A candidate face-center is detected, but it is not certain, if the rules succeed in matching human face description.

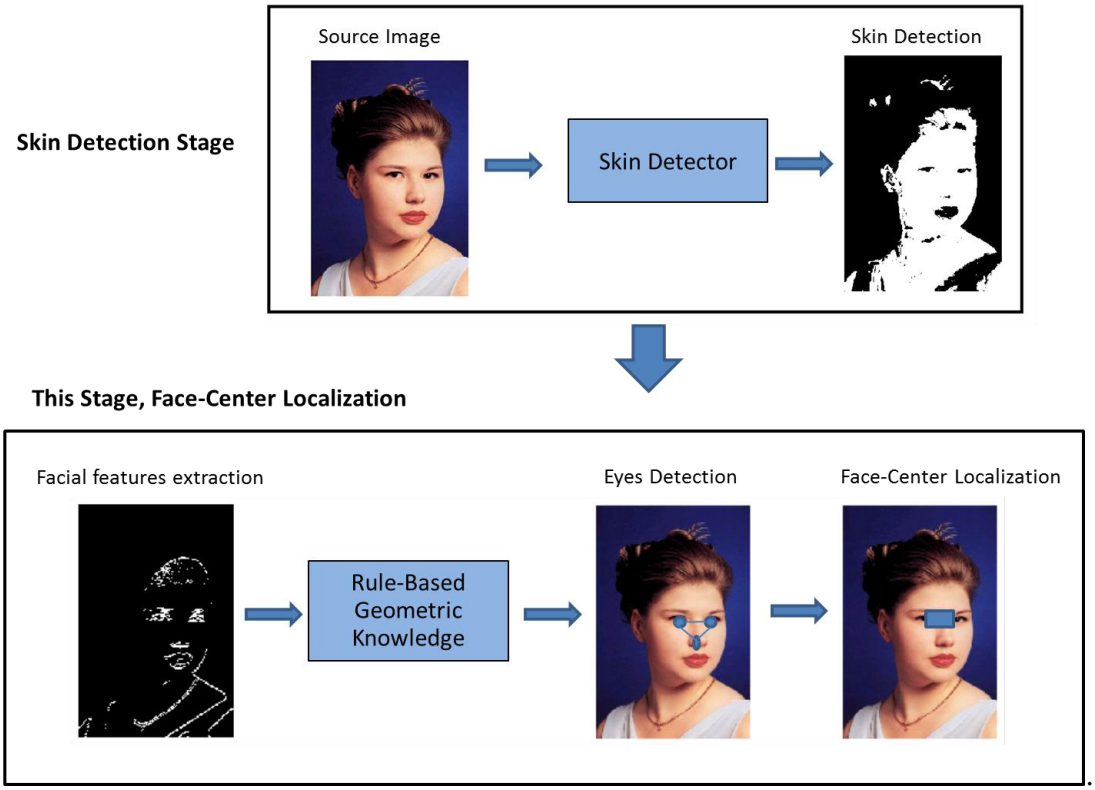


Figure 6.1: General outline of the face-center localization system.

In this research, we focus on detecting the location of eye pair. Having the location of the eyes, the rectangular area lying in between is marked as candidate's "face-center" region. Figure 6.1 shows the general outline of the face-center localization system. As shown in this figure, this step is applied to the output of skin detector. It aims at finding the position and extent of the candidate face-center region. By finding a set of candidate face-center regions, the subsequent complex classifier is supposed to work on these regions only.

The remainder of this chapter is organized as follows: Section 6.2 1describes skin segments enhancement using Convex Hull algorithm. Section 6.3 describes the facial features extraction. Human knowledge about the constitution of the face is translated into 2D geometric rules and this is presented in Section 6.4. When these rules succeed in matching human face description, a candidate face-center region is located. Generating the new search space is presented in Section 6.4.2. Experimental results and summary are presented in Section 6.5 and 6.6 respectively.

To avoid confusion, the concept of features is used to denote a piece of information which is relevant for solving a computational task related to a certain application. There is a distinction between facial features and image features. Facial features are defined as high-level entities, which are present in our faces, in accordance with our intuitive idea of the face components such as eyebrows, eyes, pupils, nose, and mouth (Zaqout 2006). Image features, on the other hand, are defined as low-level entities that can be extracted from digital images such as points, edges, curves, corners or properties of pixels or regions. Facial feature extraction (or Facial feature segmentation) is localizing the most characteristic face components (e.g. eyes, nose tip, mouth) within images that depict human faces. This step is essential for the initialization of many face processing techniques like face tracking, facial expression recognition, and face recognition.

6.2 Enhancing Skin Segmentation

In general, binary images are used to represent the result of image segmentation. Binary images play an important role in image analysis, features extraction, and object description. In most systems, binary images are stored as a logical array where each point assumes one of only two discrete values: 1 or 0. The objects in binary images are represented as a set of connected pixels of value 1 (white), while background is set to value 0 (black). The connected pixels of value 1 form a region, object, or body. Each body has at least one body coordinate system and the collection of all these points defines a volume in space. Remember that these objects or regions are called skin-maps in this research.

Although skin detection stage assumes that a suitable skin-map can be found, this may not always be true. Images seldom have very well-defined segmented skin-maps. In practice, skin-maps are irregular shapes with holes, thin gulfs, and protrusions.

This is associated with two challenges that complicate face detection:

1) For images of realistic complexity, even the most elaborate segmentation routines misclassify some pixels as foreground or background (Russ, 2007). These can be pixels

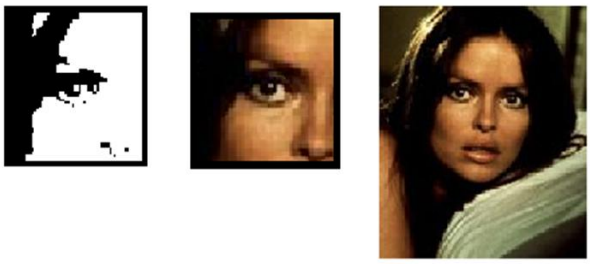
along the boundaries of regions, patches of noise within regions, or pixels that happen to share the specific property or properties used for segmentation conditions.

2) Human face may be partially detected due to variations in many random factors such as illumination, hair, etc. In other words, some facial features may be lost through image segmentation. For example, skin segmentation may produce a skin region that retains only one eye blob inside. A skin region with only one eye blob makes face detection harder. This usually occurs when an eye touches (or is partially occluded by) other object such as black hair and consequently it would be considered as part of that object (i.e. non-skin region).

Figure 6.2 show examples of different skip maps. In Figure 6.2(a), the skin maps retain most facial features. Figure 6.2(b) shows examples of missing facial features. Figure 6.2(c) shows an eye touches other objects, i.e. black hair. Before we go to the facial feature extraction step, we have to retrieve the missing facial feature; otherwise the system will fail to detect faces. The Convex Hull algorithm is used in this research for this purpose.

(a)

(b)



(c)

Figure 6.2: Skin-maps and facial features; (a) skin-maps with most facial features; (b) skin maps that miss some facial features; (c) an eye touches other objects, i.e. black hair.

6.2.1 Convex and Non-Convex Objects

In Euclidean space, a set of points S is ***convex*** if and only if for every pair of points P and Q in S, the line segment from P to Q is also in S (Apostol, 1979):

$$(\forall\, P, Q \in S \rightarrow \overline{PQ} \in S) \text{ iff } S \text{ is Convex} \tag{6.1}$$

In other words, convex is the smallest convex region enclosing a specified group of points. In two dimensions, the convex hull is found conceptually by stretching a rubber band around the points so that all of the points lie within the band. Figure 6.3(a) shows example of non-convex sets of points (or non-convex object) with a line segment outside the set (i.e. shown in red color). Figure 6.3(b) shows a convex set of points (or convex object).

Convex Hull algorithm is a powerful technique in the image processing field. It is generally used as a subsequent step after image segmentation because, in general, the results of the image segmentation operation are rarely perfect (Russ, 2007). To illustrate this point, consider the set of points shown in Figure 6.4(a). These points form irregular region boundaries (i.e. non-compact object). To get a compact body with more smooth boundaries, the Convex Hull algorithm can be helpful for this purpose. Figure 6.4(b) shows the region's boundaries after applying Convex Hull algorithm on the same set of points. As shown in this figure, it is a good tool for robust decomposition of the boundary.

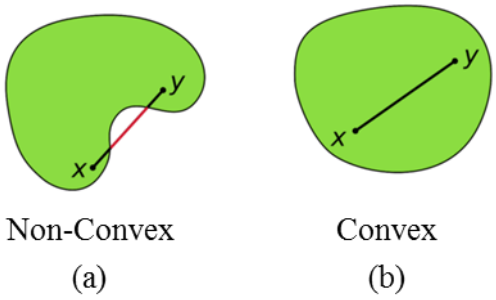


Figure 6.3: Convex and non-Convex set of points.

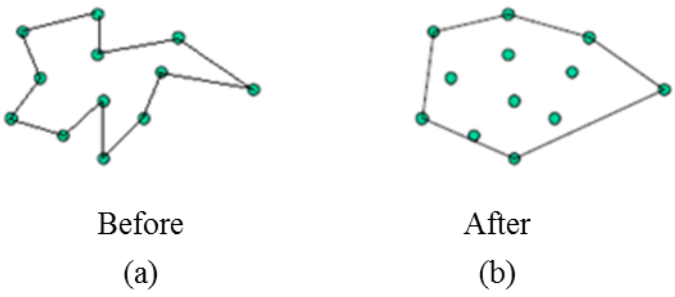


Figure 6.4: The application of Convex Hull Algorithm; (a) a set of points that form irregular region boundaries; (b) the region's boundaries after applying Convex Hull algorithm.

6.2.2 Convex Hull Algorithm

Since image segmentation techniques imply numerous imperfections, the goal of this stage is to enhance the results of image segmentation. This stage has the following goals:

- Retrieving the facial features which are missed throughout skin segmentation stage.
- Filling holes.
- Smoothing the boundaries.
- Removing gulfs.

In this work, two image-processing techniques are used to fulfill the above goals:

- Morphological closing/opening operations are used for simple border smoothing. The size of the structuring element corresponds to size of the skin-map region (i.e. area). The minimum size is 3×3 and maximum size is 9×9 square disk-shaped structuring elements. It is also used to remove small objects from the image while preserving the shape and size of larger objects.
- Convex Hull algorithm is used to approximate the elliptical shape of the face. The main advantage of this step is to retrieve misclassified parts of the human face (e.g. missing facial features).

Figure 6.5 illustrate the idea of retrieving facial features that are misclassified throughout skin detection stage. Figure 6.5(a) shows a skin-map that retains only one eye blob inside. As shown in this figure, the skin-map is non-convex and the right eye practically belongs to the background. We can retrieve the right eye by approximating the elliptical shape of the face and creating a compact region. Figure 6.5(b) shows the output of applying the Convex Hull algorithm. As shown in this figure, the region becomes convex with smooth boundaries. Then, the newly generated convex region is masked with the source grayscale image. The result of masking step is shown in Figure 6.5(c). As is evident in the example, the right eye is successfully retrieved to the human face.

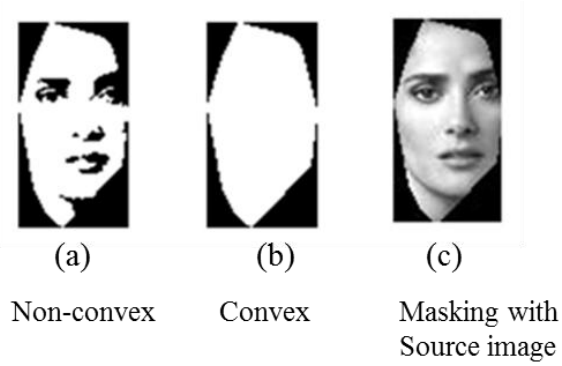


Figure 6.5: Convex Hull Algorithm; (a) original non-convex skin-map that retains only one eye blob; (b) Convex Hull algorithm is applied to approximate the elliptical shape of the face; (c) masking Convex Hull region with the source gray image.

Figure 6.6 shows various results of applying Convex Hull algorithm on real color images. The source image is shown in Figure 6.6(a); the skin-map is shown in Figure 6.6(b); Figure 6.6(c) shows the output of Convex Hull algorithm.

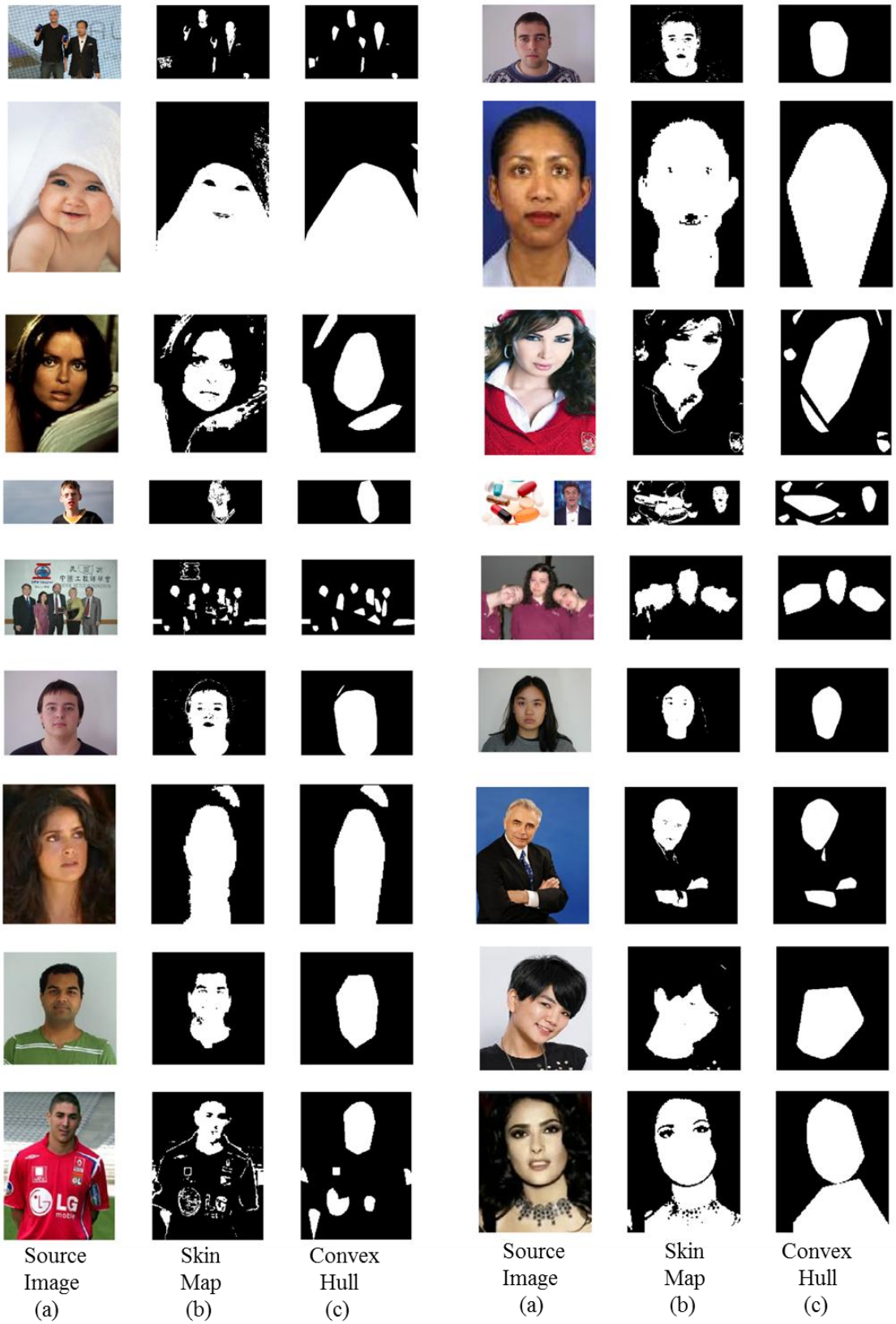


Figure 6.6: Applying Convex-Hull algorithm on skin-maps; (a) source image; (b) skin-map; (c) the output of Convex-hull algorithm.

The drawback of Convex hull algorithm is that it may attach non-skin regions along with the new generated skip regions as shown in Figure 6.7. The first two rows of this figure show the effect of hands and shoulders on the general shape of the skin segments. The third and forth rows show the effect of blond hair on skin segments. In general, facial features-based approach described in the next section overcomes this problem.

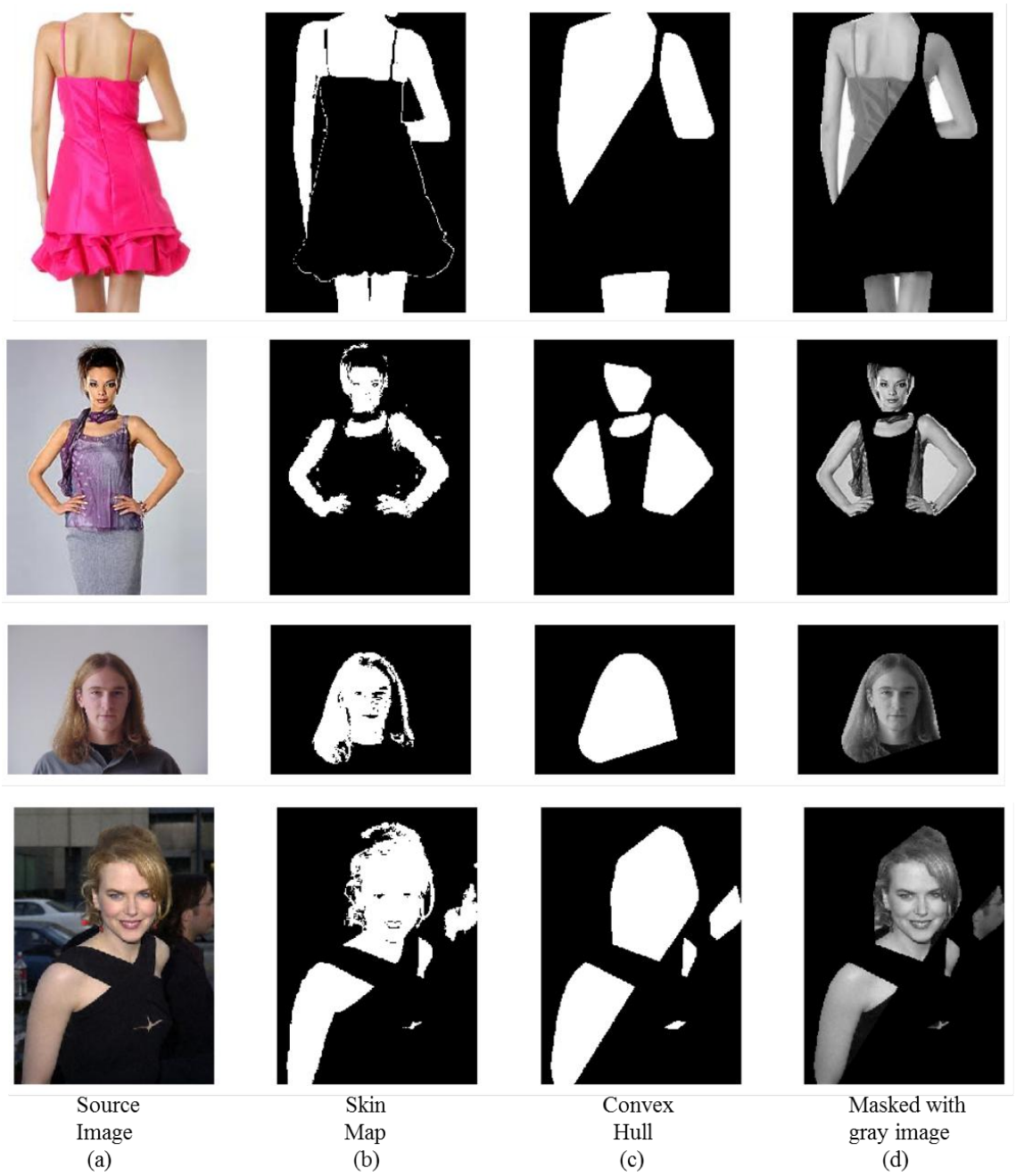


Figure 6.7: Drawbacks of Convex Hull algorithm; (a) source image; (b) skin-map; (c) Convex-hull; (c) masked with gray image.

6.3 Facial Feature Extraction

In this section we present an automatic method for facial features extraction that is used for the initialization of our rule-based geometric technique.

Extracting the facial features is equivalent to segmenting the face region into a group of entities (or objects). Based on the observation that many important facial features differ from the rest of the face because of their low brightness, we used two methods for extracting facial features: Threshold-based approach and edge-based approach. A qualitative comparison between the two methods is presented in this section.

6.3.1 Threshold-based approach

Thresholding is a segmentation technique that enjoys a significant degree of popularity especially in applications where speed is an important factor (Russ, 2007). The common form of image thresholding makes use of pixel intensity level. In this work, an image pixel with intensity value $f(x, y)$ is regarded as a facial feature if it meets the following threshold rule:

$$g(x, y) = \begin{cases} 1 & if \quad f(x, y) \le T \\ 0 & if \quad f(x, y) \ge T \end{cases} \tag{6.2}$$

where the threshold value T is calculated locally in MATLAB using Otsu's (1979) method, which is based on gray-level histogram. The output binary image g has the value of one (white) for all pixels in the input image with intensity value less than the threshold value T, and zero (black) for all others. This leads to segmenting the image in such a way that facial features such as eyes, eyebrows, nose tip, or mouth are shown as white segments on black background. Figure 6.8 shows a typical image, along with the results of threshold-based technique using Otsu's method. The source image and its skin-map are shown in Figure 6.8(a-b). The convex regions are shown in Figure 6.8(c). Masking convex regions with the original image (i.e. gray scale) is shown in Figure 6.8(d). Facial features segmentation based on thresholding approach is shown in Figure 6.8(e).

Figure 6.8: Threshold-based approach for facial features extraction; (a) source image; (b) skin detection; (c) Convex regions; (d) masking convex regions with source gray image; (e) facial features extraction using thresholding-based approach.

By considering the skin segmentation presented in the previous chapter, the system uses two levels of image segmentation. The first-level is color-based segmentation that is applied to whole image for skin detection, while the second-level image segmentation is threshold-based which is applied on the detected skin regions to extract facial features from the previously detected skin segments.

The degree of success or failure of this approach depends on the selection of an appropriate threshold. In practice, this approach is feasible in most imaging scenarios. With diversity of image types and sources, the threshold value may be falsely selected due to many factors. This will lead to facial features extraction errors. If the threshold level is too low, many facial features will be missed. If the threshold is too high, many non-facial features may be extracted as features. An obvious solution is to rely on intervention by a human operator, who can vary the threshold until acceptable results are achieved. However, this is not possible in cases where fully automatic feature extraction is required. Another method for facial features extraction is the edge-based approach as used and described in the next section.

6.3.2 Edge-based approach

Edges can be defined as locations in an image where there is sudden variation in the gray level or color of pixels (Efford, 2000). Intuitively, an edge is a set of connected pixels that lie on the boundary between two regions. *Edge detection* techniques aim to locate the edge pixels that are most likely have been generated by scene features, objects, or elements. Edge detection is a very important step in the field of image processing and computer vision, particularly in areas of feature extraction, segmentation and object recognition. By considering the smoothness of human skin (i.e. homogeneous brightness), the existence of facial features such as eyes, mouth, and nose all generate intensity edges. When we attempt to locate those features, edge detection is an essential step to take.

Although there are many edge detecting operators such as Robert, Prewitt, Laplacian, etc., Sobel operator is used because it has superior noise-suppression characteristics compared to others. Two operators to detect horizontal and vertical edges are provided by Sobel (Gonzalez & Woods, 2002) :

$$h_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \qquad h_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Based on general observation that facial features such as eyes, eyebrows and mouth always appear horizontally in the images, the horizontal operator is used to detect facial features. This will reduce the effect of vertical and diagonal edges. Figure 6.9 shows a typical image, along with the results of edge-based technique using Sobel horizontal operator. The source image and its skin-map are shown in Figure 6.9(a-b). The convex regions are shown in Figure 6.9(c). Masking convex regions with the original image (i.e. gray scale) is shown in Figure 6.9(d). Edge-based facial features extraction is shown in Figure 6.9(e). In this figure, one can notice that the skin region corresponding to the women's hand contains no facial features inside; therefore it will be excluded early.

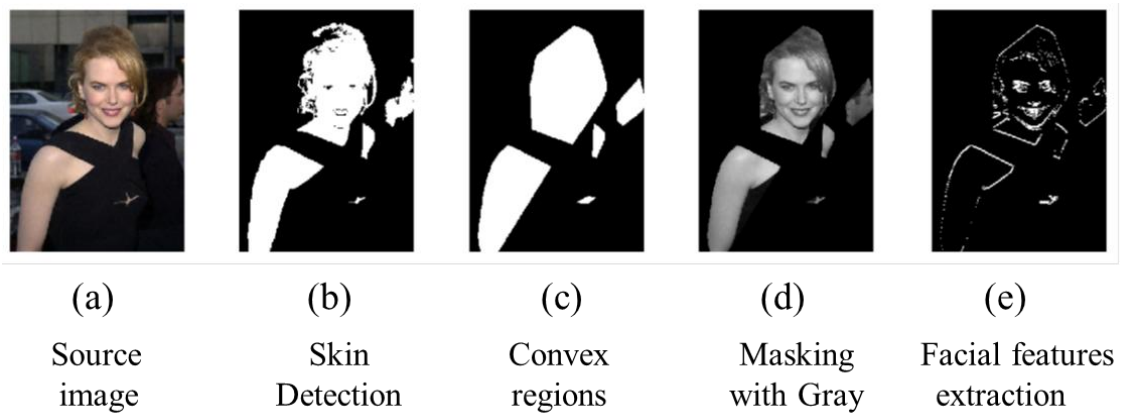


Figure 6.9: Edge-based approach for facial features extraction; (a) source image; (b) skin detection; (c) convex regions; (d) masking convex regions with source gray image; (e) facial features extraction using edge-based approach.

Figure 6.10 shows qualitative comparison between the two methods for extracting facial features: threshold-based approach and edge-based approach. In this figure, the fifth column shows threshold-based results; while the sixth column shows edge-based results. In practice, the threshold-based is simpler, faster, and more intuitive than edge-based approach, but the latter shows more detection abilities with a reduced amount of errors. With the goal of detecting all faces in the input image with minimum amount of errors, we proposed to use edge-based approach as it gives better performance.

Having processed a skin region and extracted the features, additional information about each feature such as position, area, bounding box, major diagonal, and secondary diagonal are stored and transmitted for the subsequent steps.

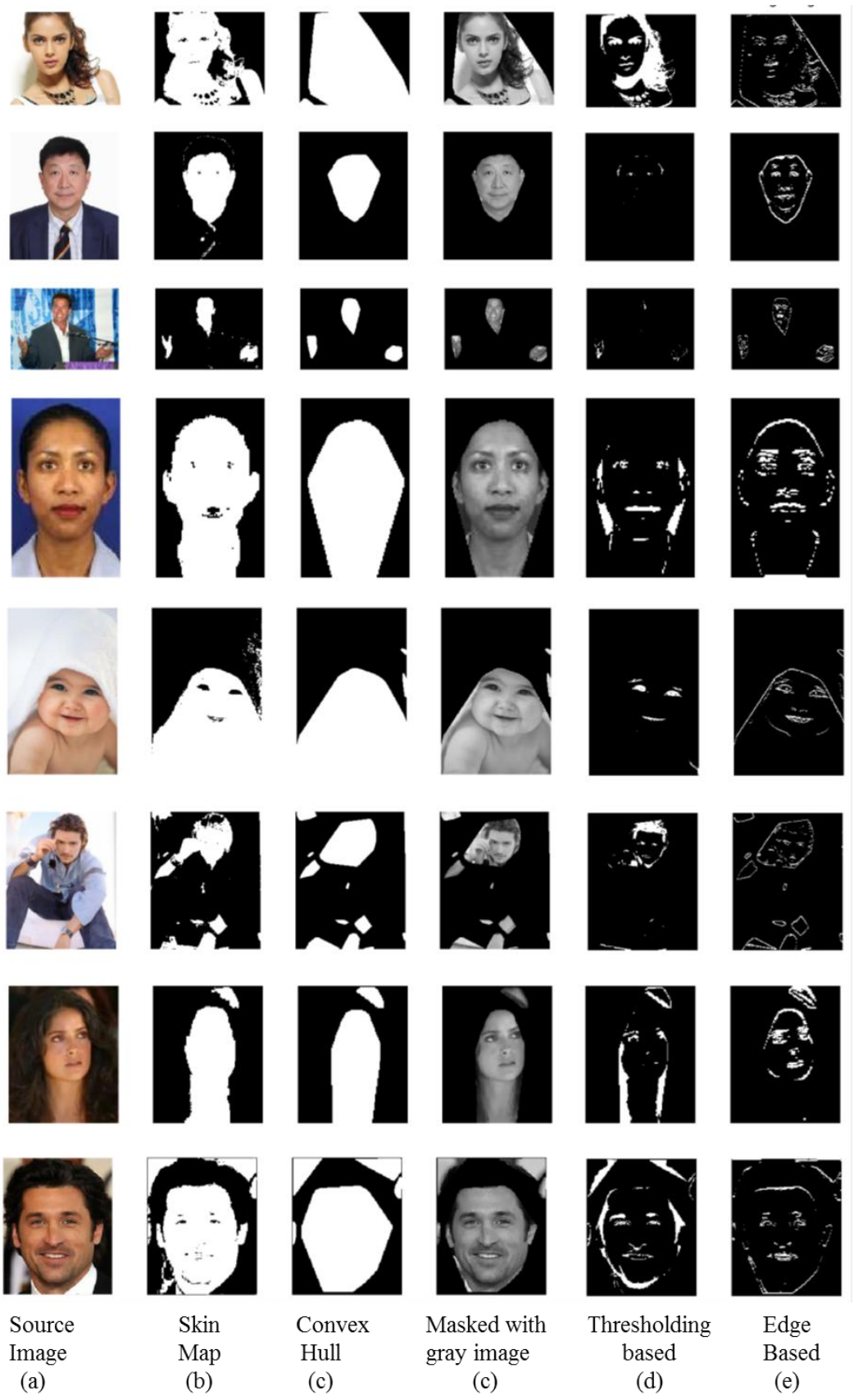


Figure 6.10: Facial feature extraction using two methods; (a) source image; (b) skin detection; (c) convex regions; (d) masked with gray image; (e) thresholding-based approach; (f) edge-based approach.

6.4 Syntactic Pattern Recognition (Rule-Based Geometrical knowledge)

In this section the focus is on the problem of recognizing face patterns by syntactic approach, which is frequently referred to as syntactic pattern recognition (Duda *et al.,* 2001). The major difference between the syntactic approach and other approaches such as statistical pattern recognition is that the former explicitly utilizes the structure of the pattern in the recognition process while other approaches deal with patterns based on quantitative numeric values of some features, thus largely ignoring interrelationship (i.e. structure) between the components of the pattern.

The problem of human face detection is difficult because of high variability in face appearance in complex images (i.e. it is non-rigid object). Therefore, using metric data alone is not enough. Snoka *et al.* (2008) stated that syntactic object description should be used whenever feature description is not enough to represent the complexity of the described object and/or when the object can be represented as a hierarchical structure consisting of simpler parts. The most important thing in syntactic approach is how it can efficiently recognize patterns using non-metric data.

The main advantage of using facial features syntactic approach is that the geometric relationships between the facial features are more invariant to changes in scale, orientation, and face pose.

The syntactic approach is based on utilization of concepts from formal language theory. Early efforts in formal language theory may be traced to the middle of 1950s by Noam Chomsky with the development of a mathematical model of grammars related to his work in natural language (Luger, 2005). The goal of his work is to develop computational grammar capable of describing natural language such as English for applications such as understanding natural language and automatic translation. The elementary properties in syntactic approach to describe objects are called primitives (or terminals). A set of rules (e.g. in a form of grammar) are used to cover the interrelationship between the primitives. A legal sentence is any string of terminal that can be derived using these rules.

The syntactic approach is widely used for pattern recognition tasks by determining whether a given pattern represents a true complete structure which can be generated by any of the rules under consideration. In an image pattern, given a set of primitives and relational descriptor we start with primitives and through the application of the descriptors, build up more complex structure. Figure 6.11 shows examples of such primitives and descriptors.

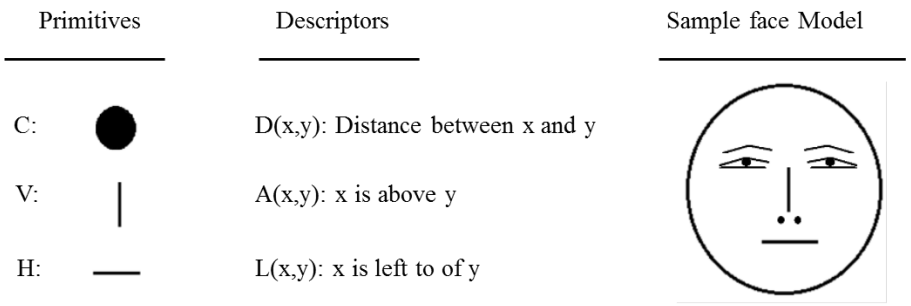


Figure 6.11: Elements of syntactic approach

In general, designing the description of primitives and relationships is based on the problem and designer experience. It requires specification of a complete and unambiguous set of rules, which have to be driven from understanding the pattern under study. However, there are some principles that are worth following (Sonka *et al.*, 2008):

- Primitives should be easily segmented from the image.
- The number of primitive types should be small.
- The primitives chosen must be able to form an appropriate object representation.
- Primitives should correspond with the significant natural elements of the object structure being described.

A problem with this approach is how to translate the human knowledge into rules that can work with all kind of images (Yang *et al.*, 2002). When using a set of rules that are detailed (i.e. more strict or specific), the system may fail to detect faces that do not pass all the rules.

6.4.1 Rule-Based Geometrical knowledge

This step considers the 2D geometrical information that encodes human knowledge of what constitutes a typical face. In ideal case, the face model as a plane is described with seven-oriented facial features, namely, two eyebrows, two eyes, two nostrils, and mouth as depicted in Figure 6.12. In general, these features have the same structure regardless of factors such as scale, location, facial expressions, and ethnicity.

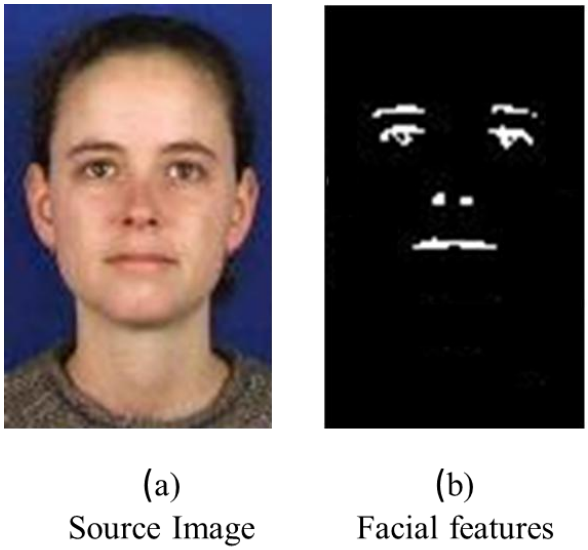


Figure 6.12: An ideal face; (a) face image; (b) the face model as a plane is described with seven-oriented facial features.

By considering the simple facial features and rules, the sample face structure is a combination of primitives. To simplify the notation, let us say:

Object 1: the left eyebrow
Object 2: the right eyebrow
Object 3: the left eye
Object 4: the right eye
Object 5: the left nostril
Object 6: the right nostril
Object 7: the mouth

We may build more complex objects starting with primitives and successively apply the rules (or descriptors). The sample pattern is used to guide the process until a total description of this pattern is obtained. Other complex objects are:

Object 8: L(1,2)
Object 9: A(8,5)
Object 10: A(8,6)

More complex object structures can be built with the objects previously generated:

Object 11: A(9,7)
Object 12: A(10,7)

But unfortunately under different viewpoints or imaging conditions, the facial features are not always present in the image or it is hard to correctly extract from the image.

In practice, we found that the face description with seven features is time consuming with many false negatives FNs due to many factors. Thus, the feature-based model needs to be decomposed into a general structure that is a common occurrence under different viewpoints and various imaging conditions.

Experimentally, with the goal to simplify the rules and made them more comprehensive in detecting more faces, only three features are considered. The two eyes are the most distinguishing features for face detection. The two eyes and one nose tip (or mouth) in the frontal view constitute an inverted triangle. Potential face regions are discovered by finding such triangles adopted by Lin and Fan (2001), Zaqout (2006), and (Lin, 2007). This structure shows flexibility to work with most image types and sources.

Face triangles are obtained by finding the combination of two eyes and nose tip (or mouth), as shown earlier in Figure 6.1. We come up with a complete set of simple and general rules to describe human face structure as shown in the next section.

6.4.2 Implementation Issues

A number of implementation issues are considered:

- **Features coordinates**: Many previous works, such as Lin and Fan (2001), Zaqout (2006); (Lin, 2007); etc., presented functions to find the center of the facial features (i.e. center of the region) such as eyes, tip of the nose, and center of mouth, as shown in Figure 6.13(a-b). These centers are used as reference points for calculating different measurements such as distance between features to apply in searching and matching the relationships among features (i.e. structure) in the 2D geometrical space. Unfortunately, in practice we find that using feature's center can mislead the search. For example, two

feature blobs may successfully pass the Euclidean distance matching rule (e.g. for finding eye pair), although these features may overlap in certain axis. In Figure 6.13(b), it is clear that some facial are overlapped in the x-axis such as <nose-tip, mouth> and <mouth, chin-edge>, although the regions' centers are not. It is known that the human eyes never overlap in x-axis (e.g. unless the face is rotated about ±90° such that one eye becomes above the other). Therefore, in this work we use the bounding box coordinates to avoid this problem. Bounding box is defined as feature's minimum and maximum limits. These limits are easy to determine by finding the pixels with the largest and smallest coordinates in the horizontal and vertical directions, as shown in Figure 6.13(c) (Russ, 2007). Bounding box coordinates can be expressed by two points, namely, Upper Left corner (Xul,Yul) and Bottom Right corner (Xbr,Ybr)..

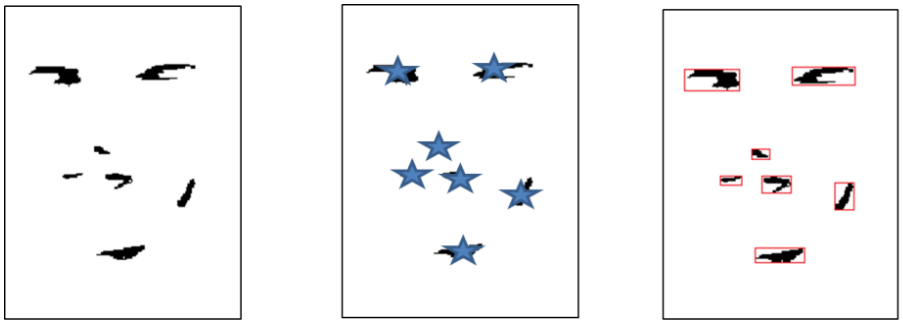


Figure 6.13: Facial features coordinates; (a) Facial Features; (b) Facial features centers; (c) facial features bounding-box.

A constraint such as:

If $X_{br,obj1} > X_{ul,obj2}$ then return; i.e. the condition does not hold (i.e. rejected)

Bounding boxes of two objects or features (obj1 and obj2) are overlapped if $X_{br,obj1} > X_{ul,obj2}$ is useful to reject overlapping features. Accordingly, a set of heuristic rules are formulated experimentally based on bounding box coordinates instead of region's center. By considering such heuristic rules, the search will be faster because when a rule

did not hold for some features, these features will be excluded from the search. Reducing the number of features speeds up the system. Please, note that the centers of objects also used in many rules where

$$X_{center} = \frac{X_{ul} + X_{br}}{2} \quad , \quad Y_{cneter} = \frac{Y_{ul} + Y_{br}}{2} \tag{6.3}$$

Any three features Obj1, Obj2, Obj3 are examined on their relative positions and inter-feature distances as shown in Figure 6.14, where

D1: Euclidean distance between Obj1 and Obj2,

D2: Euclidean distance between Obj1 and Obj3, and

D3: Euclidean distance between Obj2 and Obj3.

The Euclidean distance between two points i and j is calculated by:

$$D_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{6.4}$$

By considering Figure 6.14, the D1, D2, and D3 distances are defined as follows:

D1 = $d_{(left\_eye,\ right\_eye)}$ | $d_{(left\_eyebrow,\ right\_eyebrow)}$ | $d_{(left\_eye,\ right\_eyebrow)}$ |
$d_{(left\_eyebrow,\ right\_eye)}$

D2 = $d_{(left\_eye,\ left\_nostril)}$ | $d_{(left\_eye,\ right\_nostril)}$ | $d_{(left\_eye,\ mouth)}$ |
$d_{(left\_eyebrow,\ left\_nostril)}$ | $d_{(left\_eyebrow,\ right\_nostril)}$) | $d_{(left\_eyebrow,\ mouth)}$

D3 = $d_{(right\_eye,\ left\_nostril)}$ | $d_{(right\_eye,\ right\_nostril)}$ | $d_{(right\_eye,\ mouth)}$ |
$d_{(right\_eyebrow,\ left\_nostril)}$ | $d_{(right\_eyebrow,\ right\_nostril)}$) | $d_{(right\_eyebrow,\ mouth)}$

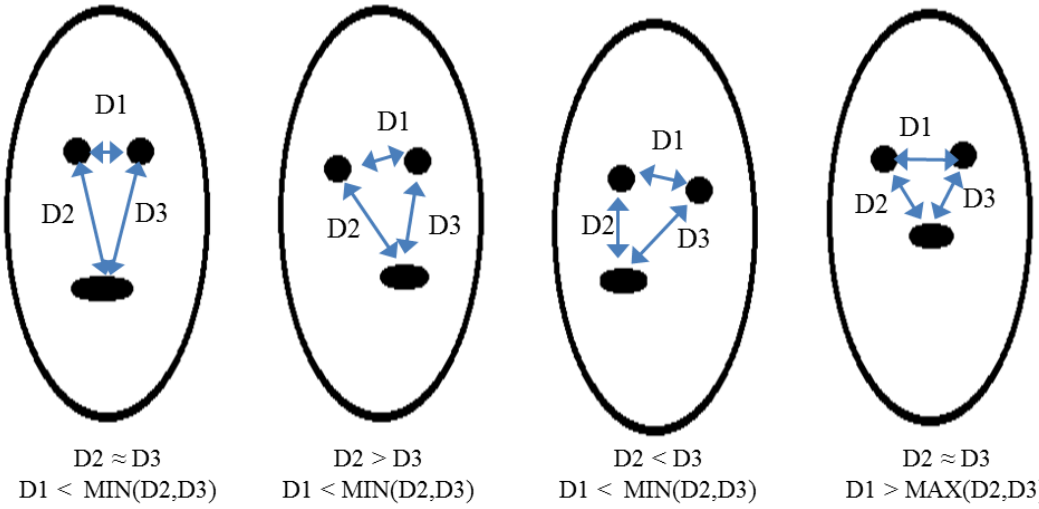


Figure 6.14: Distance between facial features.

- **Ratio measurements:** Since in real unconstrained images human face may appear in any size, we have to consider the scaling factor in describing distance between facial features (i.e. objects). Therefore, all measurements in our rules are based on ratios instead of fixed numerical values. By using ratio measurements, the rules can be applied independently to scale of the face in the image(s). By considering Figure 6.14 the ratio between D1, D2, and D3 had been found experimentally such that all true faces should pass these rules (even though it implies false positives):

  $MAX(D2,D3) \leq 1.6 \times MIN(D2,D3)$

  $MAX(D2,D3) \leq 6 \times D1$

  $MIN(D2,D3) \times 1.1 \geq D1$

  $MAX(size(eye1),SIZE(eye2)) < 1.6 \times MIN(SIZE(eye1),SIZE(eye2))$

- **Search strategy**: At run time, the search starts with primitive facial feature blobs (primitives) and successively applies the rules. The geometrical knowledge is used to guide the process until a total description of a face is obtained. In other words, the eyes are searched for by selecting each pair of features blobs in sequence as probable eyes and they are checked for orientation, location, and size constraints. Then a list of pair blobs will be generated as potential eye candidates. For each potential eye pair candidate, we start another search for nose tip or mouth. The search starts with other feature blobs by applying

different constraints that are concerned with the location of the nose tip in relation to the eyes.

- **Overlapped Detection**: Usually, there are many combinations which may satisfy the sample face model. We try to find the two eyes and one nose tip (or mouth) in the frontal view that constitutes an "*inverted triangle*". The search generates a 2D-list of size N × 3 as shown in Figure 6.15(b).

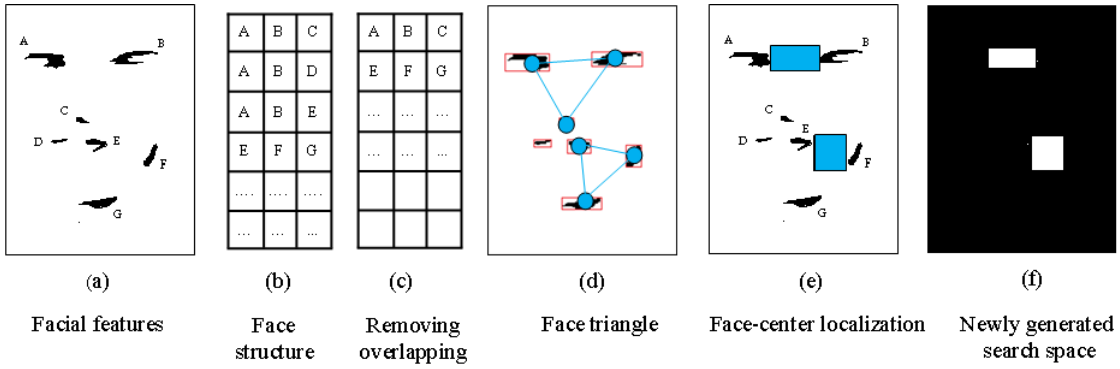


Figure 6.15: Face-center localization; (a) an example of facial features; (b) list of combinations that successfully describe face candidate structure; (c) removing overlapped detections; (d) corresponding potential faces displayed as inverted triangle; (e) face-center localization; (f) binary image representing the newly generated search space.

Each row in the list corresponds to a face candidate structure consisting of three objects: left eye, right eye and nose tip (or mouth). Usually, the combinations in the list are overlapped (e.g. <A,B,C>, <A,B,D>, <A,B,E>). Based on the simple heuristic that faces rarely overlap in images, many overlapped detections are eliminated to only one as shown in Figure 6.15(c). Figure 6.15(d) illustrates that each row in the list forms an "Inverted triangle" that corresponds to a potential face.

- **Speeding up the Search**: Many simple heuristics are used to facilitate the search process. As for example in Figure 6.15, if the search starts with blob E as left eye candidate and looking for the eye pair, all features blobs that are located at the upper-left side of E will be discarded early; these are A,C, and D. By sorting these features according to x and y

coordinates, this step will speed up the system. The same procedure is applied when looking for nose tip (i.e. the position of the candidate nose blob cannot be above the position of eye candidate blob), and so, all features blobs above the eye-pair will be discarded early.

- **Candidate Face-Center Localization (New Search Space):** The rectangular area lying between eye pair is considered a candidate "face-center" region as shown in Figure 6.15(e). This area is projected into a binary image as a set of pixels of value 1, whereas the rest of the image is set to 0 as shown in Figure 6.15(f). The figure shows two white rectangles regions (with pixel value of 1). This new generated binary image would be our new search space, instead of the whole image. Accordingly, a large amount of previously detected skin regions would be discarded.

6.5 Experimental Results

Figure 6.16 shows examples of results obtained by the face-center localization system. Figure 6.16(a) shows the source image; (b) skin detection; (c) convex-hull; (d) masked with source; (e) facial features map; (f) candidate face-center (i.e. generating new search space); (g) the candidate face-center is mapped onto the source image for illustration purpose.

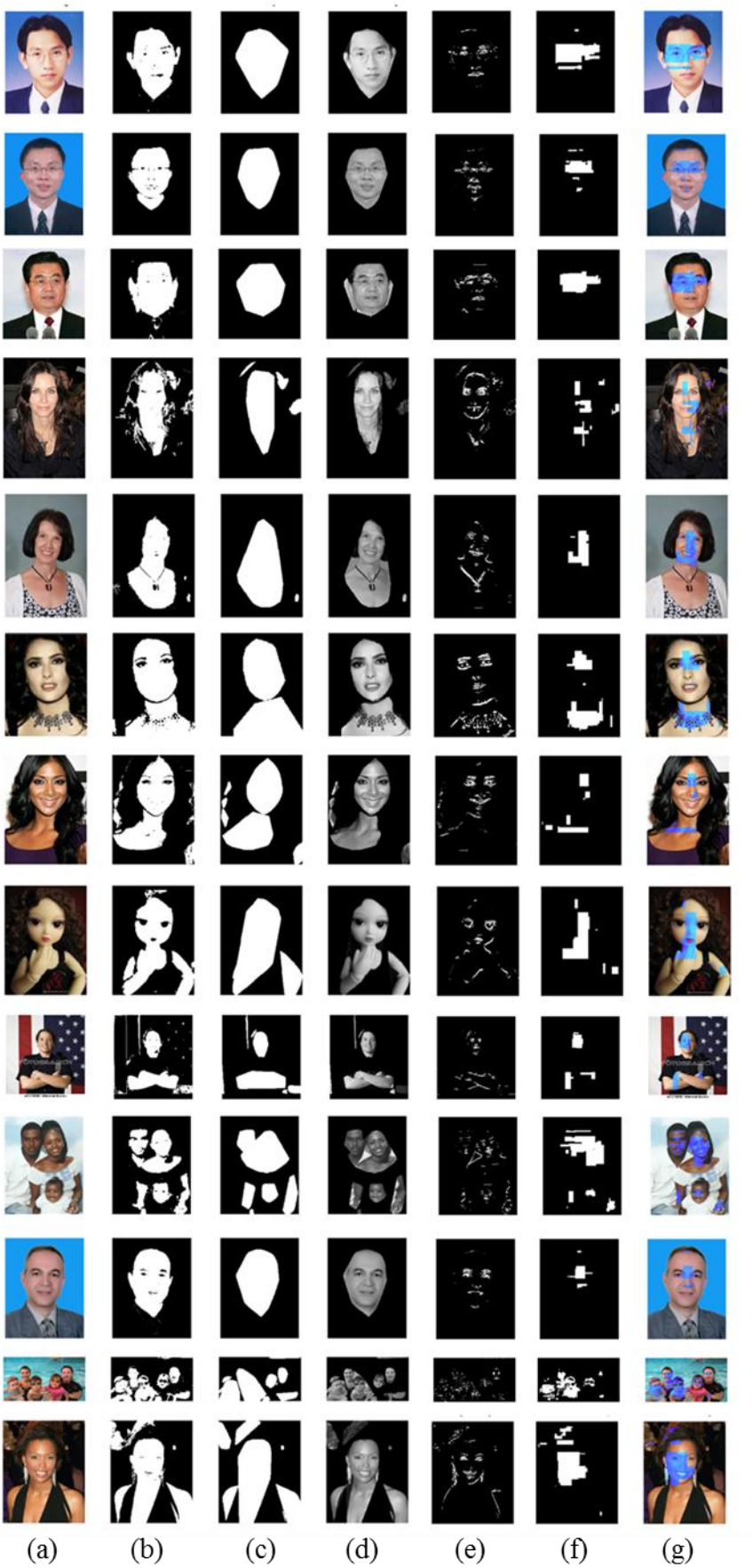


Figure 6.16: Examples obtained by the face-center localization system; (a) source image; (b) skin detection; (c) convex-hull; (d) masked with source; (e) facial features map; (f) candidate face-center (i.e. generating new search space); (g) candidate face-center is mapped onto the source image.

The main advantage of face-center localization is illustrated in Figure 6.17. This figure shows the reduction percentage in the search space. Figure 6.17(a) shows the source image; (b) the candidate face-center mapped onto the source image for illustration purpose; (c) the newly generated search space is shown as white objects at black background; (d) the reduction percentage.

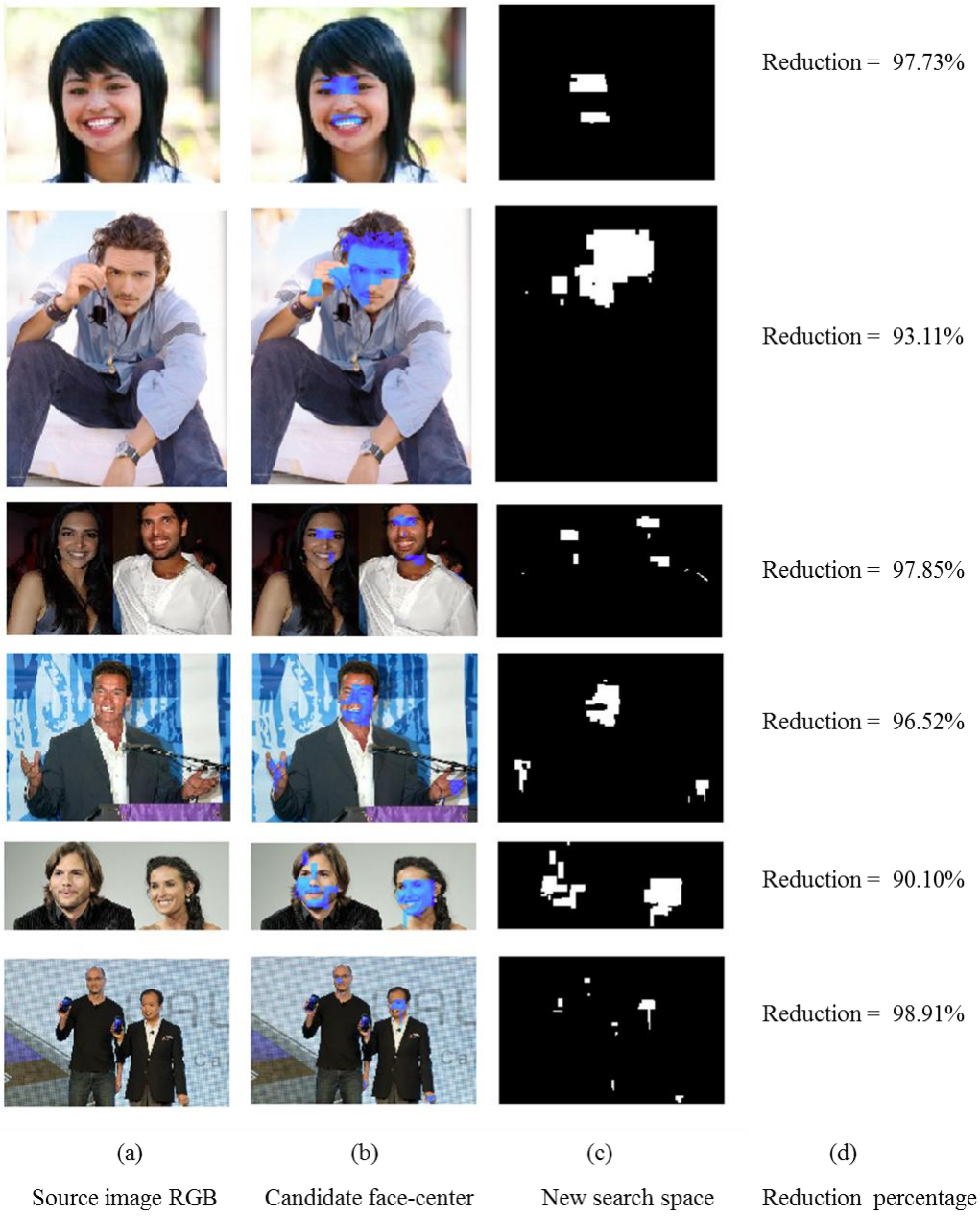


Figure 6.17: Reduction percentage in the search space;
(a) source image; (b) candidate face-center mapped at the source image. (c) new search space shown in white objects; (d) reduction percentage.

The first row of this figure shows that the reduction percentage in the search space is 97.73%, while in the second row it is 93.11%, and so on. In other words, by locating the candidate face-center regions, the subsequent complex classifier is supposed to work on these regions only. For instance consider the image in the first row; the complex classifier should process less than 3% of the total search space. By excluding most of the image's pixels (i.e. black background), the system becomes faster.

Although that processing time and the amount of the reduction in the search space depends on the image(s) contents (i.e. color, features, edge, etc.), experimental results indicate that the average reduction is about 92.32% for single face images while it is about 89.74% for complex images.

A problem with this approach is the difficulty in translating human knowledge into well-defined rules. If the rules are detailed (i.e., strict), they may fail to detect faces that do not pass all the rules. If the rules are general, they may give many FPs. Our extensive experiments on test images reveal that these rules work efficiently when facial features are easily segmented. When the faces are small, the system may miss them because the proper extraction of all facial features is difficult. This situation degrades system performance. To mitigate this problem, we used the following heuristic: When the area of a skin region is small (e.g., area $\leq$ 500 pixels, such as 25×25 pixels), geometric rules are no longer applied, that is, the region is directly passed to the next step. This process reduces the probability of missing small faces.

6.6 Summary

Images with complex backgrounds contain surfaces and objects with skin-like colors. These regions make the system more susceptible to errors and increase computation time.

This stage aims at removing false alarms caused by objects with color similar to skin color.

In this chapter, a general face-center-localization system module based on rules and geometrical measures is considered which radically reduces computation time while improving detection accuracy. The system module presented in this stage can be applied to the output of any skin

detector and returns the position and extent of the candidate face-center regions. The advantage of using rule–based approach as described in this chapter is that the geometric relationships between the facial features are more invariant to changes in scale, orientation, and face pose.

This chapter also presents a set of detailed experiments on a complex face dataset. This dataset includes faces under a very wide range of conditions including illumination, scale, pose, race, and camera variation.

This stage is clearly an effective component for face detection and it has the potential to be effective in other face processing applications such as face recognition, facial features recognition, or age estimation.

As shown in the experimental section, skin detection and rule-based approaches are designed to reject a majority of the image in order to focus subsequent processing on promising regions. They are not enough to detect faces automatically from complex images. Rather than run the risk of false detections, the system should be augmented with other processing steps such as a powerful classifier to make the final arbitration (i.e. face or non-face), which is the theme of the next chapter.

CHAPTER SEVEN

NEURAL NETWORK-BASED FACE DETECTOR

7.1 Introduction

Thus far, our approaches to the design of a face detection system have been based on locating "hot spots" (or “regions of interest”) which are likely, though not certain, to contain a face in the sense that the regions generated by these approaches are derived to restrict the search space.

The last part of the proposed system performs the face detection (i.e., classification). In this chapter, a relatively efficient appearance-based face detector based on the utilization of the machine learning concepts is presented. A computer program is said to learn from experience with respect to some tasks if its performance improves (e.g. changes in behavior) with experience and training (Mitchell, 1997). The capability of the systems to learn from experience, training examples, and other means; results in a system that exhibits efficiency in terms of speed and accuracy.

By appearance-based face detector we mean building a classifier which processes fixed-size images, and determines whether a given image corresponds to a face or not. At this step we have two-class classification problem. This can be done by using various machine learning methods such as artificial neural networks (ANNs), support vector machines (SVM), genetic algorithms (GA), decision trees (DT), etc.

ANNs are among the most effective learning methods known which have been trained to perform complex functions in various fields (Mitchell, 1997). Due to the positive impact of the pioneer approach of Rowley *et al.* (1998), many other researchers had followed this approach (Chang *et al.*, 2008; Curran *et al.*, 2005; Garcia & Delakis, 2004; Garcia & Tziritas, 1999). This chapter presents a novel ANN-based face detector called ANNFD that implies many

improvements compared with the naive implementation of face detectors for both phases: training and operation. In general, designing an efficient face detector is not an easy task and implies many pertaining problems that, if solved with suitable solutions, may highly improve the performance of the classifier. The ANNFD achieves a good trade off in terms of accuracy and speed requirements with the following points:

(i) The training face pattern is of size 15×23 pixels.

(ii) The structure of ANNFD comprises three hidden layers. Each hidden Layer is composed of a number of neurons.

(iii) The output layer contains only one neuron. The output of this neuron indicates if the presented subimage window corresponds to a face or not.

(iv) The ANNFD was trained with the backpropagation algorithm which involves supervised learning.

The remainder of this chapter is organized as follows. Section 7.2 presents the characteristics of ANNs. Data preparation is presented in Section 7.3. The design issues of ANNFD are discussed in Section 7.4. Augmenting ANNFD is presented in section 7.5. Training and operation phases of the ANNFD are presented in Sections 7.6 and 7.7 respectively. Experimental results are shown in Section 7.8. Comparison with other works and discussion are presented in Sections 7.9 and 7.10 respectively.

7.2 Why ANN-Based Face Detector

Neural network classifier is chosen due to the following main characteristics: (Haykin 2005; Pal & Mitra, 1999; Rajasekaran & Pai, 2004).

- The ANNs learn through examples. This study, when tackled with the problem of detecting human face, considers that the main advantage of using neural networks is the feasibility of training a classifier using face and non-face images to capture the inter-class variations of the human face.

- The ANNs exhibit adaptivity, that is, the ability to adjust the connections strengths to new data or information.
- The ANNs exhibit mapping capabilities, that is, they can map input patterns to their associated output patterns.
- ANNs also possess the capability to generalize. Thus, they can give the correct response even to examples that are not explicitly been shown.
- The ANNs are robust systems and are fault tolerant. They can, therefore, recall full patterns from incomplete, partial or noisy patterns as well as process information in parallel, at high speed, and in a distributed manner.
- The ANNs are rugged against failure of components (i.e. when implemented in hardware form).

7.3 Data Collection and Preparation

Training ANNFD needs a large set of training examples of face and non-face images. In this research, five datasets are used; these are FEI, CVL, FDDB, FSKTM, and JAFFE. The first four datasets had been presented through skin detection methodology (see Section 4.2.). The images from JAFFE database are not used in the previous stages of our system because these images are grayscale. The details about JAFFE database are presented hereby:

- **The JAFFE Face Database** (Japanese Female Facial Expression Database) contains 213 images of 7 facial expressions posed by Japanese female models. All images are gray scale of size 256×256 pixels and under controlled conditions. A few examples are shown in Figure 7.1. These images are used in the training phase.

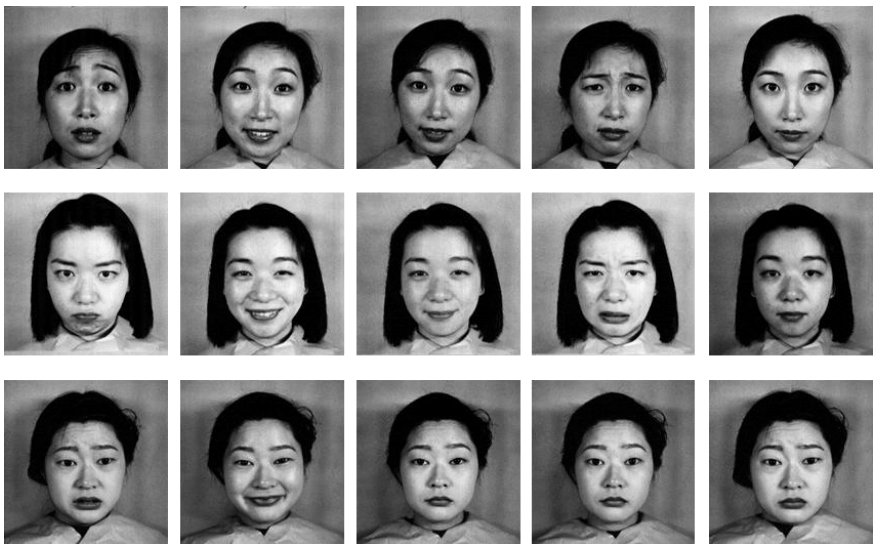


Figure 7.1: Samples of JAFFE Face Database

7.4 Design Issues of ANNFD

Design issues concerning the face model used for training (i.e. partial face pattern), alignment problem, and the preparation process of the training faces are presented.

7.4.1 Partial Face Pattern

The dependence that machine learning algorithms have on the quality of the training data is possibly one of the hardest obstacles for the successful application of those methods (Kukenys, 2010). A challenging issue affecting training data quality is the high variability in training faces that can be attributed to factors such as facial expressions and the presence or absence of structural components (e.g., glasses, beards, and moustaches). Particularly for high-dimensional problems, the input domain may be so large that all possible inputs can hardly be considered. Figure 7.2(a) shows an example of face variations for the same individual. Figure 7.2(b) shows an example of face variations among different individuals (from FSKTM dataset).

(a)
Same individual with various facial experssions

(b)
Different individuals with various expressions and the presence or absence of structural components such mustache and/or beard.

Figure 7.2: Human faces showing high variations.

The first step to improve the quality of training images is to reduce the amount of face variations between images. This would give the most compact space of face images.

To the best of our knowledge, most previous works have used whole face patterns to train the classifier, such as Figure 7.2(a). Common sizes used by researchers include 16×16, 19×19, 20×20, 24×24 and 50×50 pixels (Turk & Pentland, 1991b) (Rowley et al., 1998) (Viola & Jones, 2004) (Sung & Poggio, 1998) (Osuna et al., 1997).

Due to the general fact that the human face is oval shaped while the face images are rectangular, a number of researchers have proposed to reduce face variability using a binary mask with elliptical shape as in Figure 7.3 (Sung & Poggio, 1998). It is noticed that high variations are found at the background (outside the face oval). Eliminating (or ignoring) these regions from the training faces will ensure that the system does not wrongly introduce any unwanted background structures into the face representation (i.e. reducing images' variability).

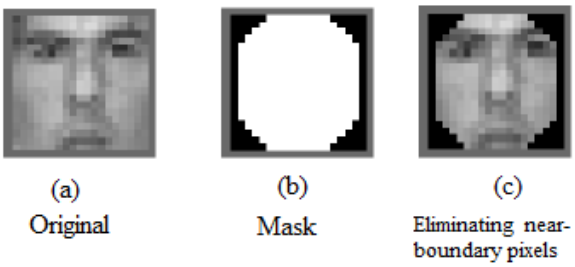


Figure 7.3: Reducing face image(s) variability by eliminating some near-boundary pixels, adopted by Sung and Poggio (1998).

In this work, standard deviation (σ) is used to figure out the most variable facial features for both cases: the same individual and then for different individuals. Suppose that we have m face images stacked in such away as shown in Figure 7.4. There are m pixels for any given coordinates (i, j), one pixel at that location for each image.

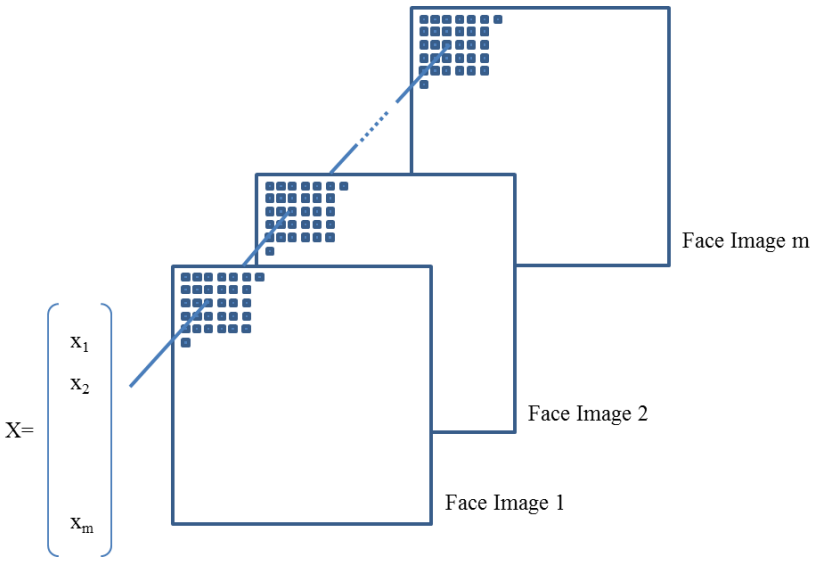


Figure 7.4: Stack of training face images of the same size.

The average and standard deviation images are calculated (Turk & Pentland, 1991a):

Step 1: Obtain face images I_1, I_2, ...,I_m (training faces) where m is the number of faces.

The face images must be *centered* and be of the same size (see Figure 7.5(a)).

Step 2: Represent every image I_i as a vector Γ_i

Step 3: Compute the average face vector Ψ (see Figure 7.5(b)):

$$\psi = \frac{1}{m}\sum_{i=1}^{m}\Gamma_i \tag{7.1}$$

Step 4: Compute the standard deviation vector (σ):

$$\sigma = \frac{1}{m}\sqrt{\sum_{i=1}^{m}(\Gamma i - \psi)\ (\Gamma i - \psi)^t} \tag{7.2}$$

Step 5: The standard deviation vector (σ) is represented as a face image (see Figure 7.5(c)).

As in Figure 7.5(c), the regions of cheeks and eyes show the lowest variations (i.e. shown as dark or black regions) compared to the mouth. Since the mouth frequently varies in shape and size due to facial expressions, the standard deviation is higher and consequently its region is shown as white. By considering this issue with other variations (such as beards and mustaches), the standard deviation becomes higher. In general, the lower part implies higher variations and of course degrades the quality of the training data.

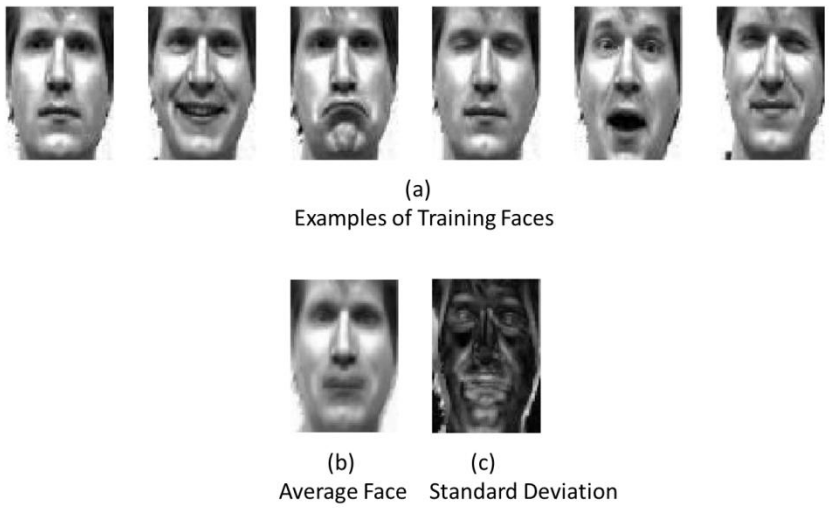


Figure 7.5: Average and Standard deviation face images for the same individual; (a) training face images; (b) average face image; (c) standard deviation face image.

Based on this conclusion, we found that using partial face pattern instead of whole face will improve the quality of the training data. The partial face pattern chosen in this study consists of eyebrows, eyes, cheeks, down to nose tip, as in Figure 7.6. The figure shows the amount of variations for the whole face pattern versus the partial face pattern. As shown in this figure,

different images of the same individual show more similarities when the partial patterns are used. Furthermore, gender differences can be greatly reduced with partial face pattern that makes the detection rate relatively similar for male and female faces. Comparing to the elliptical binary masking technique, the binary mask cannot avoid these variations.

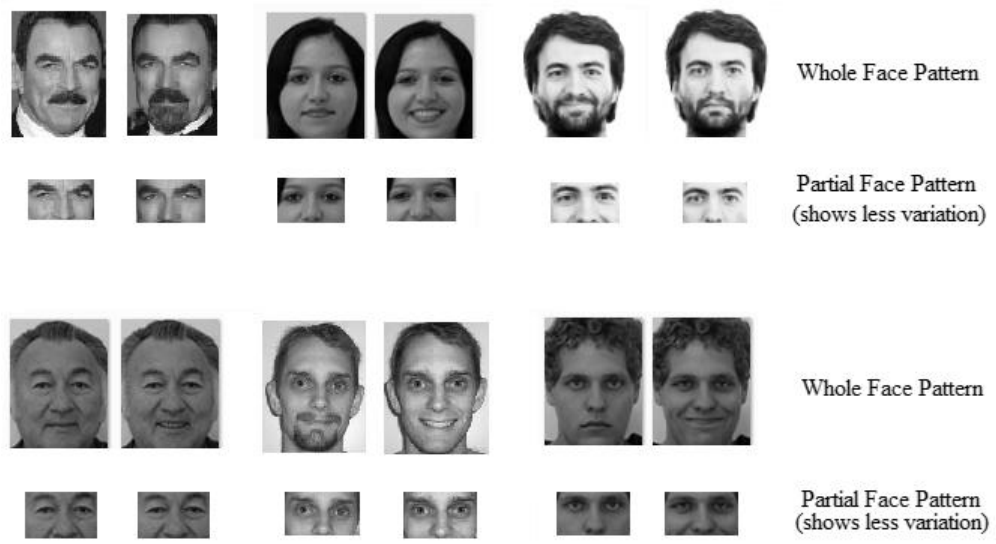


Figure 7.6: Whole face pattern versus partial face pattern.

The size of a standard face pattern that is used for training ANNFD is fixed experimentally to 15×23 pixels, as in Figure 7.7(a). Using this window size keeps the dimensionality of the face space manageably small; thereby reducing computational demands while still large enough to preserve sufficient resolution to correctly classify the images.

Figure 7.7(b) shows the ratio of distances in relation to the face center (e.g. 5N, 9N, and 11N). The usage of variable distance relations (i.e. where N is variable) makes the approach in this study relatively insensitive to the scale of the face. As N is increased repeatedly, the system can capture faces at different scales.

In practice, using partial face pattern improves the detection rate of the classifier. As the detection rate is increased, the number of false detections will increase correspondingly. It is clear that suppressing false detections is an important issue. Therefore, more features are added to the feature vector to avoid false detections and consequently improve the performance of the ANNFD (see Section 7.5).

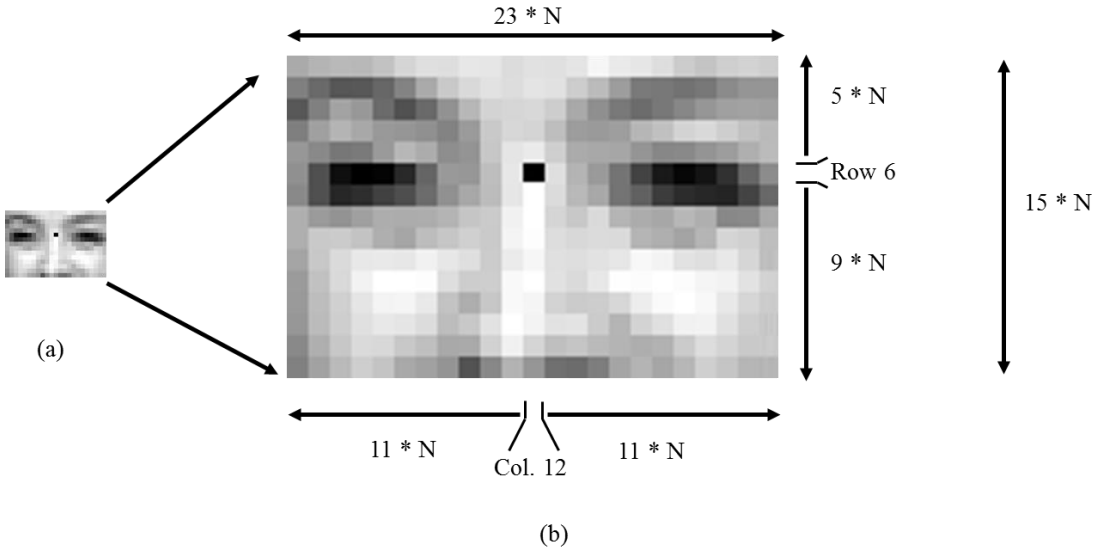


Figure 7.7: Partial face pattern; (a) of size 15×23 pixels with face center at location (6, 12); (b) the ratio of distances in relation to the face center where N is variable to capture larger faces.

7.4.2 Alignment Problem

The second step in reducing the amount of variation between training faces is to deal with the alignment problem, i.e., aligning all training faces so that the approximate positions of the facial features are the same in all images. Usually, training faces are collected from source images manually. We typically have numerous face images that can be at different scales, different rotations, or just be translated. Ideally, we would want all faces to be aligned with the least amount of variation so that we have a compact space of face images. As we crop (i.e. cut) the face images manually, this condition does not hold strictly. This variation makes it harder for the system to learn the function space of faces.

Our goal is to develop an automatic/semi-automatic method for preparing training faces instead of manual preparation with minimum amount of variation.

Many researchers have carried out this task manually while others proposed the semi-automatic approach. For instance, Rowley *et al.*(1998) handled this problem by labeling many points in each training face (i.e. eyes, tip of nose, and corners of the mouth). Then, these points were used to normalize each training face to the same scale, orientation, and position.

From our point of view this normalization causes distortion to some faces, due to the changes caused by the geometric transformations, since faces in their nature differ from one to another.

In this research, aligning of training faces was done using only one point that is referred to as the "face-center". It is defined as the intersection of two lines: the first one passes horizontally through both eyes and the other vertically upwards from the tip of the nose as in Figure 7.8. As will be shown in Section 7.4.3 the system searches for this point and starts to crop many sub-images with different sizes (a pyramid) but with the same face-center.

By considering the location of the face-center as fixed, all training faces will be prepared automatically by the system with the same relative dimension of facial features (see next section).

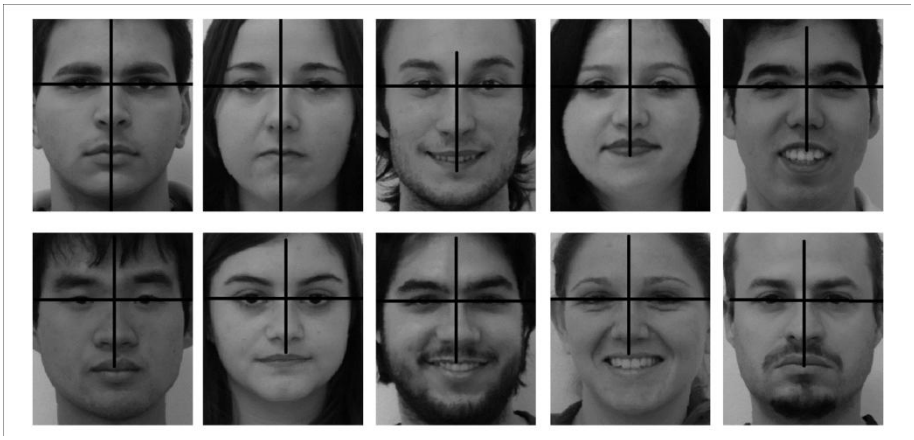


Figure 7.8: "Face-center" is labeled manually for each training face.

7.4.3 Preparing Face and Non-Face Training Examples

As mentioned before, face and non-face samples are needed to train ANNFD. In this research, a dataset consisting of 40,000 images is used. There are 20,000 positive samples of face images and the remainders are negative examples (i.e. non-face).

A novel semi-automatic method is proposed to prepare the face image. To the best of our knowledge, this is the first attempt that employs a semi-automatic method to prepare the face images without the need for normalization transformations. As mentioned before, normalization transformations may cause distortion to some faces.

In this research, the training face images are prepared as follows:

1) Faces are collected manually from images. Usually, source images contain faces of different sizes, orientations, positions, and intensities. All training faces are frontal faces rotated up to ±10°, see Figure 7.9(a).
2) The "*Face-center*" point is labeled manually for each face example. As mentioned before, it is defined as the intersection of two lines: the first one passes horizontally through both eyes and the other vertically upwards from the tip of the nose. Both lines are drawn by hand as shown in Figure 7.9(b).
3) The system automatically searches for the "face-center" point in each training face and starts cropping (or cut) from that point many sub-image windows with different sizes. The distance ratios shown in Figure 7.7(b) are used starting with N=1. Thus, the first window would be 15×23 pixels (e.g. 5.N+1+9.N=15 and 11.N+1+11.N=23). Then, N is increased repeatedly by 0.2 generating new size (real numbers are rounded to integers) and the system crops a new sub-image window at that size (i.e. 18×27, 21×32, 23×36, and so on up to the image borders); leading to a pyramid of face images (see Figure 7.9(c)). This technique will guarantee that the location of the face-center would appear at the same predetermined location in all sub-images. Then, the face center point is moved one pixel in all directions (i.e. 8-neigbor pixels) and step 3 is repeated for eight iterations. This will generate an additional eight pyramids.
4) Then, a subset of these cropped sub-images must be selected for training. We found that the number of cropped sub-images (i.e. in nine-pyramids) is high for each source image (about 200-2000 subimages depending on the image's size). Manually selecting a subset of the images to be the representative training faces is not an easy task. In practice, we found that 90-95% of these images do not resemble the proposed partial face pattern and thereby they should be excluded. So, we propose to select the best 10% of these sub-images automatically and then refine the quality of survival subimages manually. The automatic selection is done as in the following steps:
    - All images in pyramids are resized to 15×23 pixels.

- Create a "reference-template". It is the average pattern of 100 face patterns selected manually. In this work, the reference-template is used as a representative template for calculating the similarity measure.
- Compute the similarity measure between each sub-image y and the research reference-pattern x. The similarity measure is computed (Brunelli, 2009):

$$d(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2 \qquad (7.3)$$

- A small value of $d(x, y)$ is indicative of pattern similarity.
- Sort all sub-images in ascending order based on the similarity measure.
- Select the top 10% of the sub-images from the list. The remaining 90% of images are rejected.

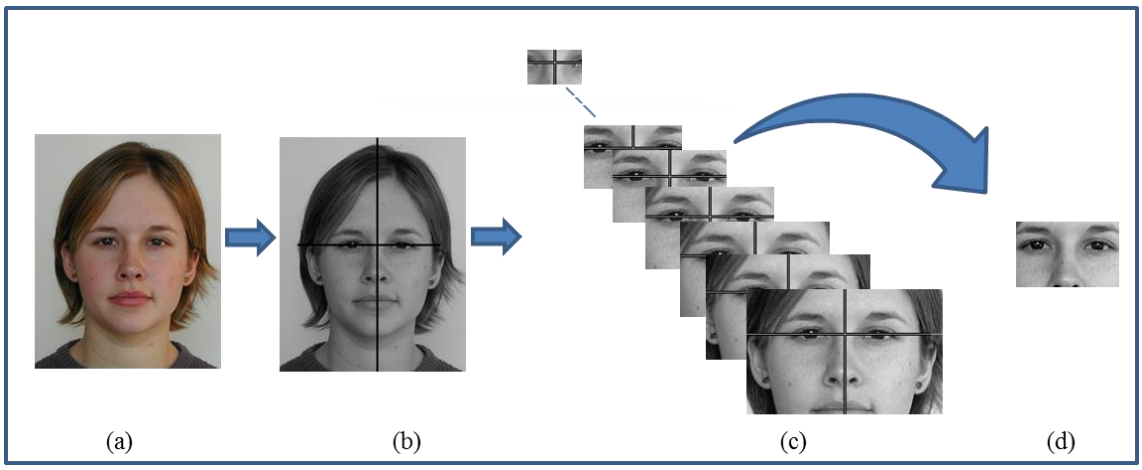


Figure 7.9: Preparation of training faces; (a) source image. (b) "Face-center" labeled manually (c) pyramid of face images with the same face center are generated automatically; (d) the selected training face.

- Furthermore, to refine the quality of the training images, some of these survived images are rejected based on user judgment (subjective inspection). The survived images must contain eyebrows, eyes, cheeks, down to the nose tip with minimum window boundary as in Figure 7.9(d). In general, it was found that about 5% to7% of the pyramid sub-images were kept as training faces for this research.
- To expand the range of intensities in the images and improve contrast, histogram equalization is used.

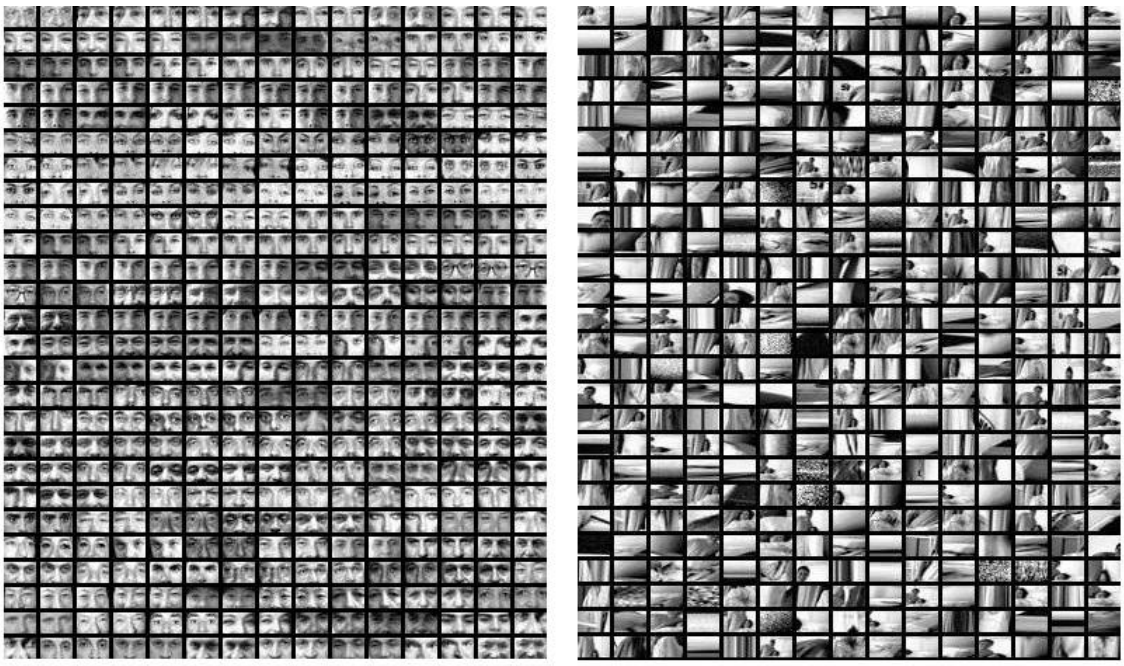
Figure 7.10(a) shows samples of face images used for training the ANNFD. All face images are of fixed size 15×23 pixels.

The next step is to prepare "non-face" patterns, which is a challenging task in neural network training. Obtaining a representative sample for the non-face images is difficult. The problem is how to describe and characterize a "non-face" pattern. We can say that any sub-image that does not contain a face can be characterized as a non-face image. This makes the space of non-face images very large compared to the face images. Instead of collecting the non-face images by hand, non-face images are collected automatically as follows:

- A set of source images are selected at random.
- Demolish all faces to obtain images which contain no faces.
- Run the system to crop 20,000 sub-images at random window sizes and locations.
- Resize sub-images to 15 × 23 pixels.
- Apply the preprocessing step to improve contrast and normalize intensities.
- Images with natural patterns that resemble faces are discarded.

Figure 7.10(b) shows samples of non-face images used for training the neural network. All non-face images are of fixed size 15×23 pixels.

The collected images are divided into three subsets: training, testing, and validation (i.e. 14,000, 13,000, and 13000 images respectively). The training set is used first to train the ANNFD. The test set is used to measure the performance of the ANNFD while the validation set is used to further refine its structure.



(a) (b)

Figure 7.10: Examples of face and non-face samples used for training ANNFD; (a) face images (i.e. partial face pattern); (b) Non-face images;

7.5 Augmenting ANNFD

When solving a pattern recognition problem, the ultimate objective is to design a system which will classify unknown patterns with the lowest possible probability of misrecognition (Garcia & Tziritas, 1999). Although ANN-based face detectors can be trained and tested to show perfect performance on the training dataset, they may lead to poor performance when applied on real images. The misclassification errors may be due to the following reasons:

1) In traditional ANN-based classifiers, each pixel's intensity x_i represents the i^{th} descriptor in the feature vector as in Figure 7.11. Although researchers usually use small size images for training and classification, it is still an extremely high dimensional space. For instance, in this research we used a small face pattern of size 15×23 pixels (i.e. feature vector of 345 inputs); but actually each image pixel is represented internally by one byte and described by a grayscale intensity value between 0 (Black) and 255 (White). Therefore, we have 256^{345} possible combinations of gray values. With such enormously high dimensional

space, it is computationally expensive and hence feature vector based on pixels intensities alone is not enough to construct an efficient classifier.

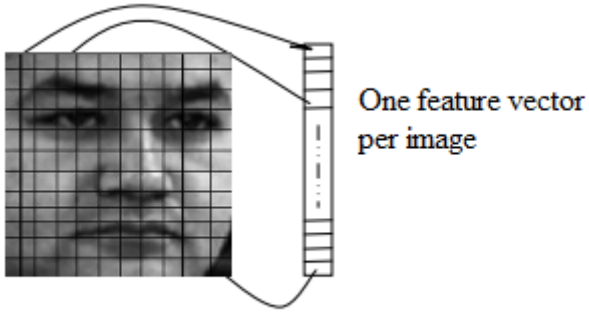


Figure 7.11: Pixel-based feature vector, adopted from (Sarfraz, Hellwich, & Riaz, 2010).

2) Training ANNs for large-scale problem is challenging due to the fact that it needs a huge amount of storage. Although the quality of the representative training data can be improved using Bootstrapping algorithm, the limitation of memory resources makes training harder. It is really hard to get representative samples that define face space in such high dimensional space. While the size of the training data (i.e. face and non-face images) used for the face detection problem may be large from the machine learning perspective, it typically presents only a very small subset of all the possible inputs that can represent a valid human face (Kukenys, 2010). Even with an assumption of preparing a huge amount of sample images to train the NN in batches (e.g. 150,000,000 images), Rowley (2000) stated that: by the time we reach the end of 150,000,000 examples, it will have "forgotten" (lost track) the characteristics of the first images.
3) In most cases, the network will only approximate the desired function because a neural network is built from a set of standard functions, and even for an optimal set of weights the approximation error is not zero (Krose & Smagt, 1996).
4) The huge number of sub-image windows is implied in a single source image (i.e. enormous search space). For instance, consider arbitrary input image with medium resolution of 700×750 pixels. The size of the search space using sliding window technique for the first pass (i.e. scale=1) is about a half million sub-image windows (i.e. 700×750 = 525,000). All

these sub-images must be classified. Even with an assumption of a perfect performance classifier, the probability of false detection remains high due to the fact that the number of false detections is proportional to the size of search space.

Thus, it is a good idea to augment the ANNFD by a suitable means to improve the performance. In this research, two methods are proposed for this purpose:

- **Constraints (or Heuristics):** proper constraints should be placed on the network so that the number of the patterns is reduced as much as possible without reducing its computational accuracy.
- **Augmenting the Feature Vector:** if we do not augment the input vector with adequate features, even the most sophisticated classifiers may fail to accomplish the classification task (Hadid, Pietikainen, & Ahonen, 2004). Therefore, it is important to derive new features which:
    - Discriminate between classes well while tolerating inter-class variations.
    - Can be easily extracted from raw face images in order to allow for fast processing.
    - Lie in a low dimensional space (short vector length) in order to avoid computational cost.

The remainder of this section presents the X-Y-Reliefs constraints in Section 7.5.1, and augmenting the feature vector with texture features and wavelet coefficients in Sections 7.5.2 and 7.5.3 respectively.

7.5.1 X-Y-Reliefs Constraints

This section presents the X-Y-Reliefs constraints which are used as a fast initial classifier placed on the network to discard a lot of negative patterns (or non-face windows). They are computed for each sub-image window by processing the horizontal and vertical profiles, as such is related to the work of Baskan *et al.* (2002) and Kotropoulos and Pitas (1997). However, unlike these works which assume a single face canonical image(s) as a precondition, we generalize the idea

for all kinds of images. The above-mentioned works applied the idea of horizontal and vertical profiles on the entire image which makes it of limited use because it is hard to locate the facial features when dealing with complex background as shown in Figure 7.12(a). Furthermore, this method has difficulty detecting faces in multiple-faces images as in Figure 7.12(b), adopted from Yang et al. (2002).

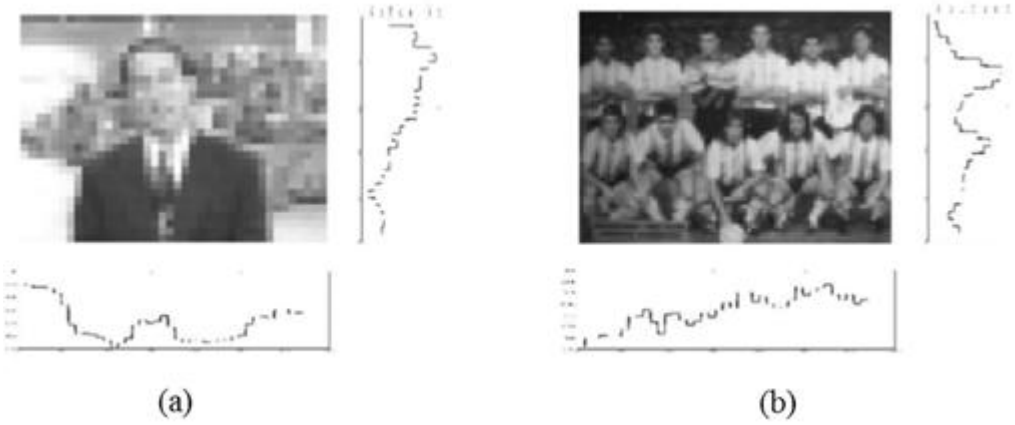


Figure 7.12: X-Y-Reliefs method; (a) complex background; (b) multi-faces image.

On the other hand, our scheme does not have this draw back because the horizontal and vertical profiles are applied on each sub-image window rather than the whole image. Thus, the idea of horizontal and vertical profiles becomes practical. The Y-Relief is obtained by summing all pixel intensities in each row. Similarly, X-Relief is obtained by summing up all pixel intensities in each column. The horizontal and vertical projections are respectively defined as follows:

$$HI(x) = \sum_{y=1}^{n} f(x, y), \qquad VI(y) = \sum_{x=1}^{m} f(x, y) \tag{7.4}$$

Then, it is easy to locate facial features by detecting the local maximum (or minimum) and first abrupt transition. As shown in Figure 7.13, each facial feature generates a maximum in Y-Relief and has specific X-Relief characteristics. In this research, the first feature E_1 relies on the property that the region of the eyes is often darker than the cheeks, as in Figure 7.13(a). The second feature E_2 relies on the property that the nose tip is often darker than the cheeks. As in Figure 7.13(b), the third feature E_3 relies on the property that the bridge of the nose is often lighter than the eyes.

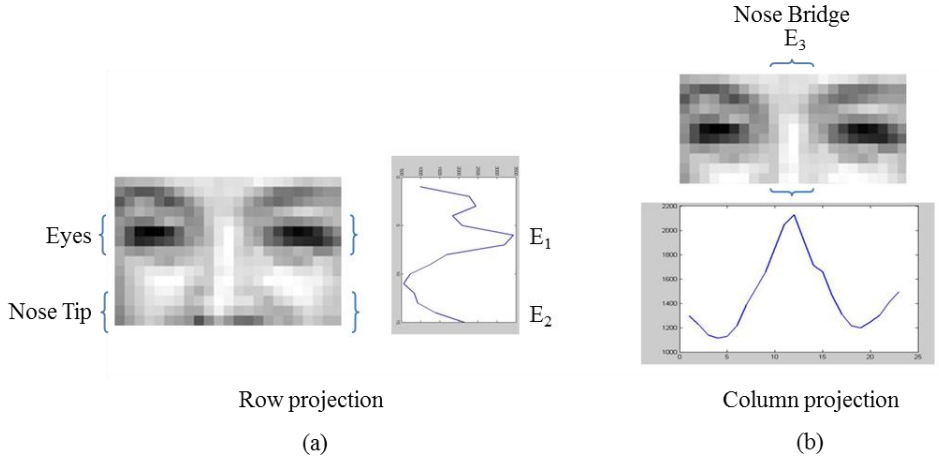


Figure 7.13: The X-Y-Reliefs constraints; (a) E_1 and E_2 features rely on the property that the eyes and the nose tip are darker than the cheeks; (b) E_3 feature relies on the property that the bridge of the nose is lighter than the eyes.

In this research, the X-Y-Reliefs classifier is adjusted experimentally with three constraints C1, C2, C3 as follows:

$$C1: 3 \le E_1 \le 7$$

$$C2: E_2 \ge 13$$

$$C3: 9 \le E_3 \le 13$$

This classifier is capable of rejecting many non-face windows early, while most of the face windows are passed successfully. Those sub-windows which are not rejected by this initial classifier are processed by the ANN-based classifier. When a subimage window is rejected, no further processing is performed.

7.5.2 Texture Features

An important approach for describing a region is to calculate a set of metrics designed to quantify its texture contents. Texture refers to the local variation in brightness from one pixel to the next or within a small region (Russ, 2007). In general, texture gives us information about the spatial arrangement of color or intensities in an image or a selected region of an image. By considering the importance of texture descriptors, a set of texture descriptors are proposed to augment the feature vector. In this case, not only the intensities of pixels x_i are considered in

the feature vector but also the content of neighboring pixels (i.e. spatial relationship between them).

In this research, statistical properties for describing region texture are used. Mean (*m*) (or average darkness), standard deviation (σ) (or amount of variation in a region), and smoothness (*r*) (measures the relative smoothness of the intensity in a region), adapted from Gonzalez et al. (2007), are defined as follows:

$$m = \sum_{i=0}^{L-1} z_i P(z_i) \tag{7.5}$$

$$\sigma = \sqrt{\sum_{i=0}^{L-1} (z_i - m)^2 P(z_i)} \tag{7.6}$$

$$r = 1 - 1/(1 + \sigma^2) \tag{7.7}$$

where z_i is a random variable indicating intensity, $P(z_i)$ is the histogram of the intensity levels, and L is the number of possible intensity levels.

In this work, the face pattern is divided into six regions, namely: two eyes, two cheeks, nose forehead, and nose tip as shown in the first row of Figure 7.14 . As human faces have a distinct texture and the same type of facial features have the same brightness, a set of texture descriptors are calculated from these regions. The proposed descriptors M_1-M_9 are, similar to those of (Sinha, 2002), based on a collection of statistical properties and several pair-wise ordinal contrast relationships across facial regions as shown in Figure 7.14. These are:

M_1: darkness ratio of left cheek to left eye < left-cheek, left-eye>.

M_2: darkness ratio of right cheek to right eye < right-cheek, right -eye >.

M_3: darkness ratio of nose bridge to left eye < nose-bridge, left-eye>.

M_4: darkness ratio of nose bridge to right eye < nose-bridge, right-eye>.

M_5: darkness ratio of left cheek to nose-tip < left-cheek, nose-tip >.

M_6: darkness ratio of right cheek to nose-tip < right-cheek, nose-tip >.

M_7: darkness ratio of left cheek to right-cheek < left-cheek, right-cheek >.

M_8: smoothness of the left cheek region.

M_9: smoothness the right cheek region.

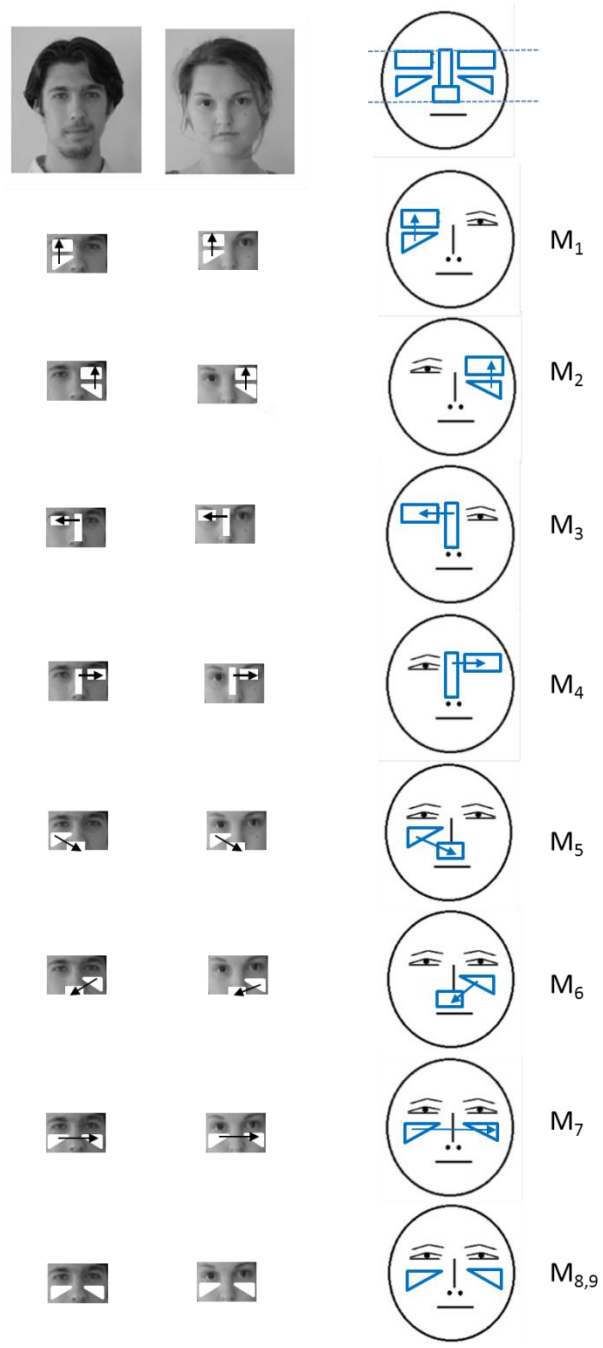


Figure 7.14: Texture descriptors and their pair-wise ordinal relationships.

The average brightness of the left eye is always less than that of the left cheek, irrespective of the lighting conditions. The average darkness of the two regions may change, but the relationship between the average darkness of the < left cheek, left-eye> pair is invariant under

lighting changes. By using the ratios of darkness between selected facial regions, the method becomes relatively insensitive to the illumination. The figure also shows other descriptors such as the smoothness of regions. By augmenting the image-based feature vector x_i with the new texture descriptors M_i, the classifier's discriminating power is improved.

Three sets of experiments were performed to evaluate the classifier performance:

- A conventional image-based feature vector (pixels intensities only).
- Texture-based feature vector only.
- Integrated image and texture feature vector.

The comparison among performance evaluations is as in Table 7.1 which shows the effectiveness of the classifier that is based on integrating both image and its texture features.

Table 7.1: Classifier performance with different feature vectors.

Feature vector of ANN-Based Classifier	Accuracy
Image-based feature vector (pixels intensities only).	94.30%
Texture-based feature vector only.	93.50%
Integrating image and texture feature vector	98.79%

7.5.3 Wavelet Coefficients

Another way is used to create a solid and meaningful feature vector in which not only the intensities of pixels are considered but also features set based on *Wavelet coefficients*.

Unlike the statistical texture features, which are based on spatial domain only, the Wavelet packet decomposition provides a multi-scale analysis of the image in the form of coefficient matrices with a spatial and a frequency decomposition of the image at the same time (Garcia & Tziritas, 1999; Gonzalez *et al.*, 2007).

The wavelet transform has a particular advantage over the earlier transforms, such as Fourier transform, in that it does not make the assumption that the image repeats on all directions, and so is able to have better space-frequency localization (Russ & Christian Russ, 2008). Fourier analysis is not local in space, but is local in Frequency (Starck, Murtagh, & Bijaoui, 1998).

In the last two decades, wavelets have become very popular and have been applied to many applications such as image compression, de-noising, texture discrimination, detection of faults, signal matching, surface reconstruction from contours, and features extraction. A brief background on wavelets is already presented in Section 2.3.6. The discrete wavelet transform DWT, which is based on sub-band coding is found to yield a fast computation of the wavelet transform. The signal is passed through two complementary filters and emerges as two signals, Approximation and Details. This is called *decomposition* or *analysis*. The components can be assembled back into the original signal without loss of information. This process is called *reconstruction* or *synthesis* (Mohideen, Perumal, & Sathik, 2008).

The extension to the 2-D case is usually performed by applying these two filters separately in the two directions. In the 2D image, an N level decomposition can be performed resulting in 3N+1 different frequency bands namely, LL, LH, HL and HH. The low-frequency component LL is the approximation image and the three high-frequency components along different directions *horizontal* (HL), *vertical* (LH) and *diagonal* (HH) are detail components. The low-frequency component contains the average information and most of the energy of the image, while the high-frequency components contain the details of the images.

The image is first decomposed into four sub-bands denoting LL1, LH1, HL1 and HH1. However, LH1, HL1 and HH1 contain the finest scale detailed wavelet coefficients, that is, the higher frequency detailed information while LL1 contains the low frequency component. The wavelet transform is then applied by further decomposing LL1 into LL2, LH2, HL2 and HH2; and if the process is repeated *n* times, we can obtain the sub-band LLn through t-scale level wavelet transform. For instance, 2D-DWT with three levels decomposition for sample face image is shown in Figure 7.15(a-b). The upper left block is a smoothed approximation of the original image; it comes from three iterations of the lowpass filter with down-sampling. The other subbands contain details at various scales.

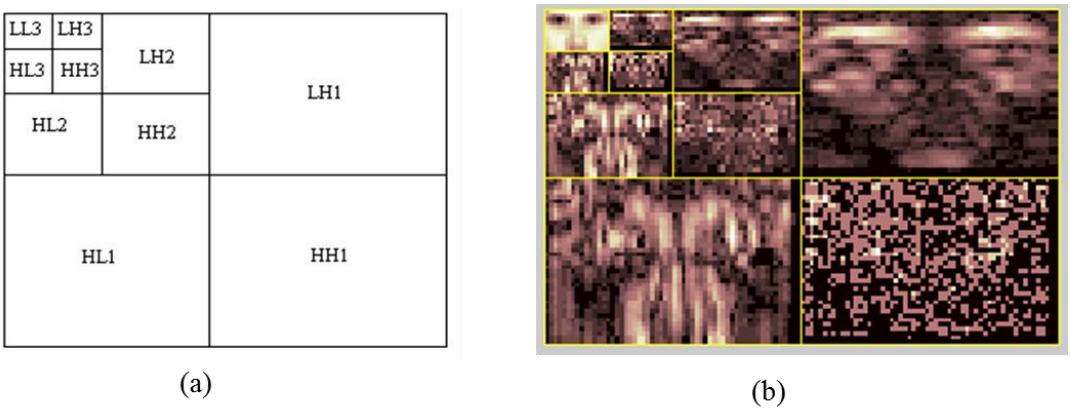


Figure 7.15: 2D-DWT with three levels decomposition for sample face image.

One of the major issues when using DWT is choosing a suitable wavelet filter. The Daubechies family of wavelet filters is perhaps the most popular because of its simple interpolation schema (Daubechies, 1992; Russ & Christian Russ, 2008). In this work, Daubechies-4 (db4) is used with 6-levels decomposition to extract 27 features of wavelet coefficients from gray scale face images to keep the number of features reasonable. The Daubechies-4 treats 4-pixel-wide region using weights of $((1+\sqrt{3})/4,\quad (3+\sqrt{3})/4,\quad (3-\sqrt{3})/4,\quad -(3+\sqrt{3})/4)$ (Russ & Christian Russ, 2008). Accordingly, 3×9 wavelet coefficients are extracted from each training image (i.e. nine for the horizontal, nine for the vertical, and nine for diagonal detail coefficients). Table 1 shows an example of extracted of wavelet coefficients (H, V, D) from the grayscale face images at level 6 for 15×23 image.

Table 7.2: Sample of extracted wavelet coefficients at level 6 for (15×23) image

Horizontal detail coefficients (H)	**Vertical detail coefficients (V)**	**Diagonal detail coefficients (D)**
44.7483557024718	98.8846565244710	-6.21032513421047
48.2315348067625	-106.689718321882	-7.55305529705817
23.7310353574212	7.80506179740944	13.7633804312676
-1280.39621435390	110.967128684840	24.5399432972958
-1230.83202824022	-28.6670225091717	266.188575582763
-315.303211620572	-82.3001061756694	-290.728518880059
1235.64785865143	197.640037500937	-18.3296181630853
1182.60049343345	895.461087166552	-258.635520285705
291.572176263149	-1093.10112466749	276.965138448791

Three sets of experiments were performed to evaluate the performance of the classifier:

- A conventional image-based feature vector (pixels intensities only).
- Wavelet coefficients-based feature vector only.
- Integrated image and wavelet coefficients feature vector.

The performance of the classifier for the three experiments is shown in Table 7.3. In this table, one can notice that using wavelet coefficient-based feature vector shows poor performance (i.e. accuracy =67.40%). Furthermore, integrating these coefficients with pixels intensities (i.e. face image) does not improve the performance of the classifier. Therefore, we found that texture features based on statistical descriptors (see Section 7.5.2) show better results than wavelet coefficients. Accordingly, the statistical descriptors were adopted in this work as they show better results.

Table 7.3: Classifier performance with different feature vectors.

Feature vector	Accuracy
Image-based feature vector (pixels intensities only).	94.30%
Wavelet coefficients-based feature vector only.	67.40%
Integrated image and wavelet coefficients feature vector.	81.33%

7.6 ANNFD Training Phase

As mentioned before, there are several types of architecture for ANNs and various learning mechanisms (Section 2.3.4). In this research, the Feed Forward Back propagation Neural Network (FFBP-NN) is used. The structure is called feed forward as the information flow is in one direction along connecting pathways, from the input layer to the final output layer (i.e. no backward connections exist between neurons from different layers). Back propagation involves supervised learning and it depends on the delta rule to adjust the weight value between two units. Through the delta rule, the adjustment of a weight can be defined by computing the difference between what we got and the desired output. Figure 7.16 shows the process of FFBP supervised learning used for training ANNFD.

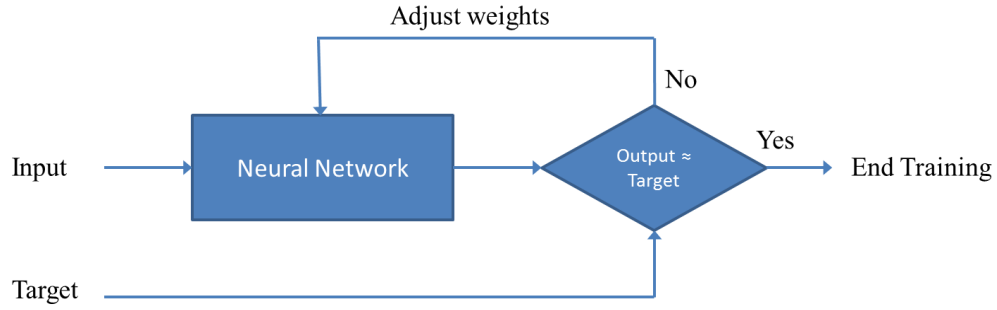


Figure 7.16: The process of FFBP supervised learning used for training ANNFD.

The learning task involves classifying gray images to face or non-face images. In total, about 40,000 gray scale images were prepared (see Section 7.4.3), each with a resolution of 15×23, with each image pixel described by a grayscale intensity value between 0 (Black) and 255 (White). Our target is that the ANNFD can learn to high accuracy from these images.

To train ANNs for the face detection problem, a number of design choices must be made (Mitchell, 1997). We summarize these choices as follows: ANNFD input, ANNFD output, ANNFD structure, and ANNFD learning parameters with the details described in the subsequent sections.

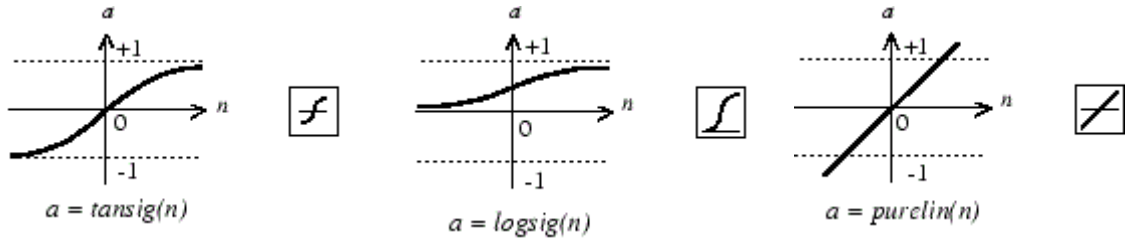
7.6.1 ANNFD Input

Given that the ANNFD input is to be some representation of the image, a key design choice is how to encode this image. For example, we could preprocess the image to extract edges, regions of uniform intensities, or other local image features, and then input these features to the network. A difficulty with this design option is that it would lead to a variable number of features (e.g. edges) per image, whereas the ANN has a fixed number of input units. The design option is to encode the image as a fixed number of input units. In this research, 345 pixels' intensities plus 9 texture features are used as the input feature vector. Texture features are used to improve the classifier performance as was illustrated in Section 7.5.2.

7.6.2 ANNFD Output

A further design choice here is "what should be the target values for the output units". As shown in Figure 7.17, many shapes for the transfer function in an ANN can be used. Each function may fit a specific application. In the research, the hyperbolic tangent sigmoid transfer function (i.e. $tansig(n)$) is used in all neurons of the hidden layers as in Figure 7.17(a). This function is selected because it is a robust differentiable function over an infinite range. The output layer contains one neuron. The output neuron is estimated using the linear transfer function (i.e. $purelin(n)$) as in Figure 7.17(c). This neuron generates an output ranging from -1

to +1, indicating the presence or absence of a face, respectively. This is the natural choice which fits the result expected by the ANNFD. If the presented subimage window matches perfectly the face pattern, the ANNFD desired output is +1, else the output is less down to -1 (i.e. non-face).



Hyperbolic tangent sigmoid (a) | Log-sigmoid (b) | Linear transfer function (c)

Figure 7.17: Three main types of transfer function in ANN.

7.6.3 ANNFD Structure

ANNFD structure implies determining how many layers to include in the network structure and how many neurons in each layer. It is common to use one, two, or three hidden layers. Using more layers makes the processing time longer.

To find the best network hidden layers structure (the input and output layers are already determined), a number of experiments should be performed. In this research, more than 3000 different NN structures had been tested. The particular choices of number of neurons per layer were driven empirically through a trial and error process in which the number of neurons was increased until a significant detection rate is achieved (i.e. from 1 to 15 neurons per layer). The network's weights are initialized with random values. Then, face and non-face images are repeatedly presented as input with the corresponding desired targets. The output is compared with the desired target, followed by error measurement and weights adjustment until the minimum error rate is reached.

Examples of the performance of different versions of network structures are shown in Table 7.4 tested on 13,000 test images (i.e. 5,000 face images and 8,000 non-face images). The first three columns show the number of neurons in each hidden layer. The last three columns show the detection performance of the networks in terms of Accuracy, FN, and FP. The best detection rate of the ANNFD is 98.97% with network hidden layers structure (6-4-2) that is six neurons in the first hidden layer, four neurons in the second hidden layer and two neurons in the third hidden layer.

Table 7.4: Detection and error rates for different versions of the network structures. The highlighted rows show the detection accuracy above 98.5%.

Number of neurons in			Accuracy	FN	FP
First Hidden Layer	Second Hidden Layer	Third Hidden Layer			
…	…	…	…	…	…
1	1	0	96.13876	302	200
1	2	0	95.44650	407	185
2	1	0	94.47735	436	282
…	…	…	…	…	…
3	2	1	97.1308	84	289
3	2	2	96.1462	170	331
3	3	1	98.2231	91	140
3	3	2	98.2923	55	167
3	3	3	97.13077	140	233
…	…	…	…	…	…
…	…	…	…	…	…
5	4	2	97.03077	125	261
5	4	3	98.38462	75	135
5	5	2	98.84615	32	118
5	5	3	98.79231	54	103
…	…	…	…	…	…
5	6	2	97.93077	61	208
5	6	3	98.72308	51	115
…	…	…	…	…	…
5	7	2	61.40769	27	4990
5	7	3	98.83077	41	111
…	…	…	…	…	…
6	4	2	98.97692	43	90
6	4	3	83.86154	1484	614
…	…	…	…	…	…
6	5	2	97.77692	112	177
6	5	3	97.90001	58	215
6	5	4	98.91538	39	102
6	5	5	93.38462	197	663
…	…	…	…	…	…
8	2	2	97.61538	94	216
8	2	3	97.60769	112	199
8	2	4	97.96923	93	171
8	2	5	98.48462	88	109
8	2	6	97.95385	72	194
…	…	…	…	…	…
14	14	10	77.12484	1349	1625
14	14	11	85.92416	1260	570
14	14	12	77.14791	1695	1276
…	…	…	…	…	…

The receiver operator characteristic ROC curve is also used in this research to assess the performance of several different classifiers. The ROC curve shows the trade-off between true positives and false positives using different threshold values. Figure 7.18 shows the ROC curves of different versions of the NN structures that have been tested in this research. These are:

Net-01: The hidden layers structure is (14-14-08) with an accuracy =69.956%.

Net-02: The hidden layers structure is (06-04-03) with an accuracy =83.861%.

Net-03: The hidden layers structure is (08-08-08) with an accuracy =76.146%.

Net-04: The hidden layers structure is (06-04-02) with an accuracy =98.976%.

As shown in this figure, Net-04 shows the best performance that agrees with the detection rates shown in table 7.3. Therefore, the weights of this NN structure are saved in the system secondary storage. When the ANNFD is initialized it just reloads the saved weights.

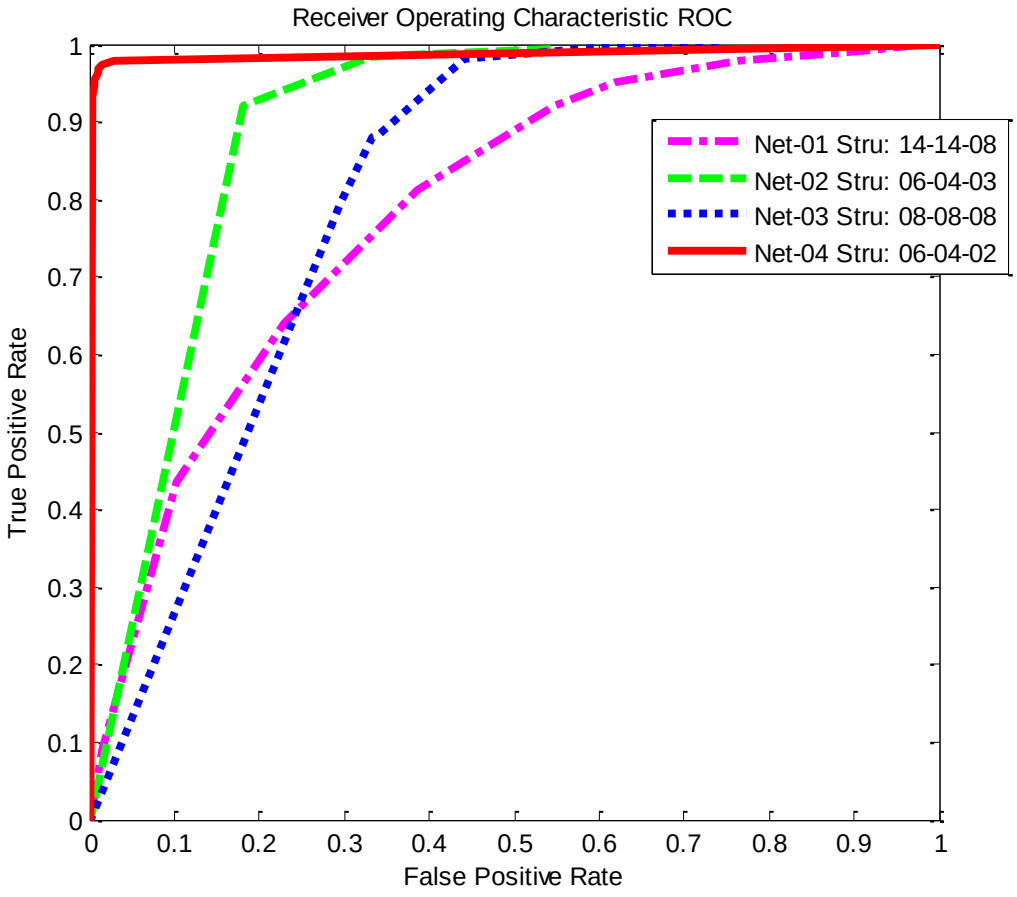


Figure 7.18: ROC curve of different versions of NN structures.

7.6.4 ANNFD learning parameters

We had to choose some parameters in advance for learning experiments such as *learning rate* η and *momentum* α. The role of the learning rate η is to moderate the degree to which weights are changed at each step. It is usually set to some small value (e.g. 0.1) and sometimes made to decay as the number of weight-tuning iterations increases (Mitchell, 1997). When the learning rate η is small, the backpropagation algorithm proceeds slowly. The effect of the *momentum* α is that if the basic delta rule would be consistently pushing a weight in the same direction, then it gradually gathers "momentum" in that direction.

Lower values for each of them require longer training time. If these values are both high, training fails to converge to a network with acceptable error over the training set. In this research the values of η =0.05 and α =0.5 are used to train the ANNFD.

7.7 ANNFD Operation Phase – Classification Stage

In this work, the search space has been greatly reduced using pre-processing steps as shown in the preceding chapters. Therefore, the operation of the ANNFD does not need to search over the entire image. Only candidate face-center regions are searched. These regions are already detected and labeled in a separate binary image called the “Face-center” or “new search space image”. Therefore, the classification process will examine an image location only if the corresponding location in the new search space image is ON.

The ANNFD operation phase consists of four steps: the cropper, histogram equalizer, texture-analyzer, and ANN-based classifier.

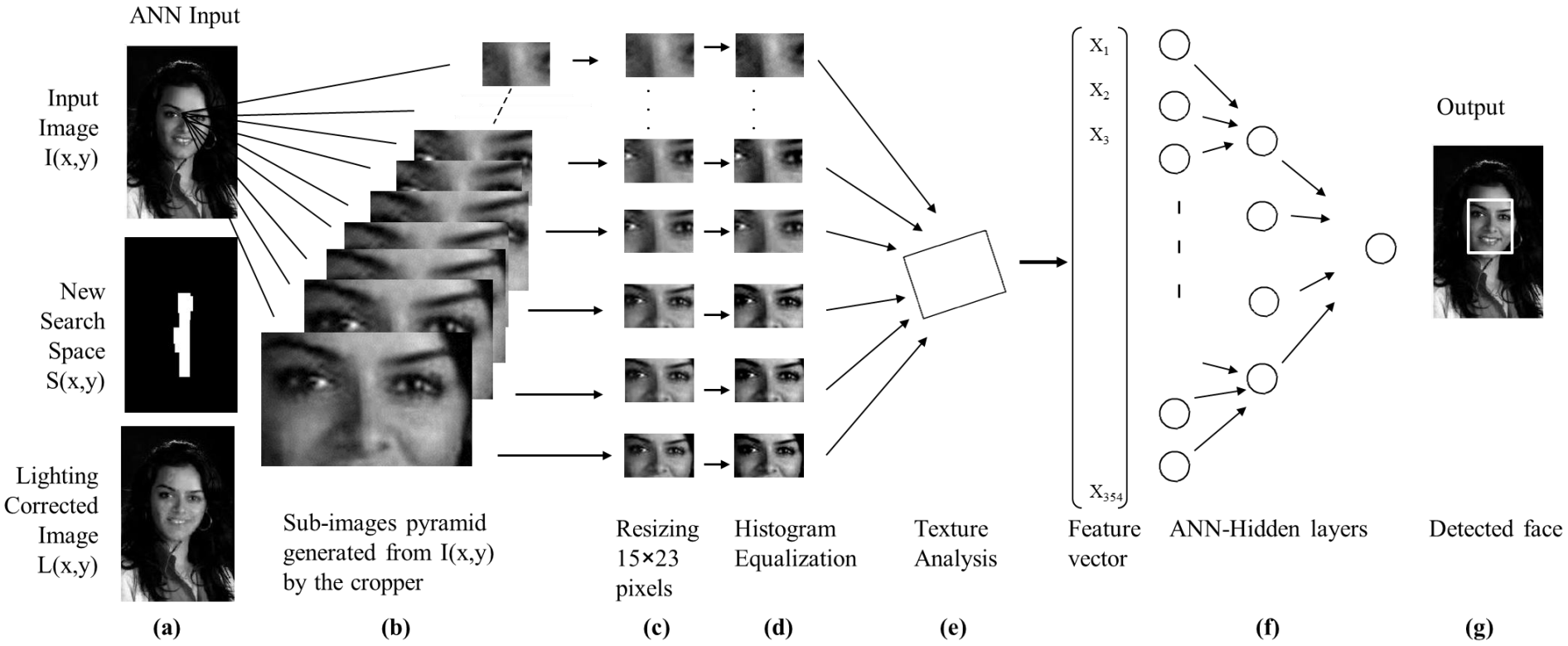


Figure 7.19: The classification stage of ANNFD; (a) The ANN input; (b) sub-images pyramid; (c) resizing to 15×23 pixels; (d) Histogram equalization; (e) texture analysis; (f) the hidden layers of neural network classifier; (g) the ANNFD output image.

- **The cropper:** receives as input three images of the same size: the input image I(x,y) that is gray scale, the new search space S(x,y) that is binary image, and the lighting-corrected image L(x,y) that is gray scale, as shown in Figure 7.19(a). The S(x,y) binary image is used to guide the search process. When a pixel S(x,y) is ON then its corresponding location at I(x,y) is considered as a candidate face-center otherwise it is ignored (not included in the search). The function of the cropper is to crop a sub-images pyramid from I(x,y). The window size starts with N=1. As in the training phase, the first window would be 15×23 pixels (i.e. 5.N+1+9.N=15 and 11.N+1+11.N=23). Then, N is increased repeatedly by 0.2 generating new size (real numbers are rounded to integers) and crops a new sub-image window at that size for each step (i.e. 18×27, 21×32, 23×36,..., up to image boundaries); leading to a pyramid of sub-image windows as shown in Figure 7.19(b). The main advantage of using proportional measures relative to face-center location is to ensure that the location of the face centers would appear at the same predetermined location in all sub-images. The cropper then resizes each sub-image to the standard pattern size of 15×23 pixels as shown in Figure 7.19 (c). The sub-images are passed to the next step, that is, the histogram equalizer. To detect faces at different locations, the cropper moves to every location which is ON in S(x,y) and starts cropping at that location from I(x,y) generating another pyramid.

- **Histogram equalizer**: improves the contrast by expanding the range of intensities in the sub-image window as shown in Figure 7.19(d). The pre-processed sub-image is then passed to the texture-analyzer.

- **Texture-Analyzer**: calculates 9 texture descriptors from each image and attach these descriptors as input features to be passed to the ANN-Based face detector (also called classifier). These features are useful to locate facial features such as eyes, cheek, and nose tip as well as describe the relationship between them. This will enhance the feature vector with more efficient features.

- **ANN-Based face detector**: receives as input 15×23 pixels sub-image window plus 9 texture descriptors. The function of the face detector is to decide whether the sub-image contains a face or not. It generates an output ranging from -1 to +1, signifying the presence or absence of a face, respectively. When the system detects a face image, its location is saved; otherwise the classier checks the corresponding sub-image at the lighting-corrected image L(x,y). Using lighting-corrected image makes the system more efficient to deal with illumination variations. All the detected sub-image windows need further processing such as removing overlapped detections (see Section 7.7.2).

Unlike other works, there is no need to perform lighting correction at classification time. This step is highly important because it affects the speed of the system. In general, other works pre-process each sub-image window by applying the lighting correction process before passing it to the face detector. This imposes a high computational cost due to the fact that this process may be repeated millions of times (for each sliding window) and thus causing serious delay to the classifier operation, while the technique proposed by this research does not have this drawback. In this research, lighting correction is done prior to the classification stage as described in Chapter 5.

7.7.1 Speed-up the System

In this section, we discuss some methods to improve the speed of the system:

- The cropper operation: the system repeatedly reduces the source image size instead of increasing the size of the sub-image window (i.e. pyramid). This reduces the computations required for re-sampling and consequently speed-up the system substantially.
- Texture descriptors: are used to improve the performance of the ANNFD but at the expense of high computational cost compared to the other steps. Viola and Jones (2004) introduced the "Integral Image" which is another form of image representation used to quickly calculate responses of a set of image-based features. To speed up the system, all

calculations of texture descriptors could be done in advance using the same idea of Viola and Jones.

7.7.2 Eliminating Overlapped Detections

Practically, ANNFD may produce multiple positive detections for a face, because the same face can be detected at multiple scales and at several nearby positions. This leads to multiple overlapped detections. Figure 7.20 shows examples of the raw output of the ANNFD that contain a number of overlapped detections. It is clear that, merging and eliminating these overlapped detections should be one of the functions of the system.

In practice, it is noticed that the location of face centers of these detections tends to cluster about a typical nearby region. Based on simple heuristic that faces rarely overlap in images, we have proposed to eliminate such detections based on measures of distance between pattern vectors. As mentioned before, a representative face template called a reference-template is used in this research (Section 7.4.3). At this step suppose that we have a set of positive detections, the system computes a set of Euclidean distances $d_i(x)$ from each member of the positive detections to the reference-template of this research (i.e. $d_i(x), d_2(x), ..., d_n(x)$). The i^{th} detection is the best face detection, when it yields the smallest Euclidean distance:

$$d_i(x) < d_j(x) \qquad i, j = 1,2,...,n; \qquad i \neq j \tag{7.8}$$

The system keeps the location of such detection and eliminates the other detections. Figure 7.21 shows an example of eliminating overlapped detections. The raw output of the ANNFD is shown in Figure 7.21(a); there are multi-overlapped detections for the same face at very close locations. The result of eliminating the overlapped detections is shown in Figure 7.21(b).

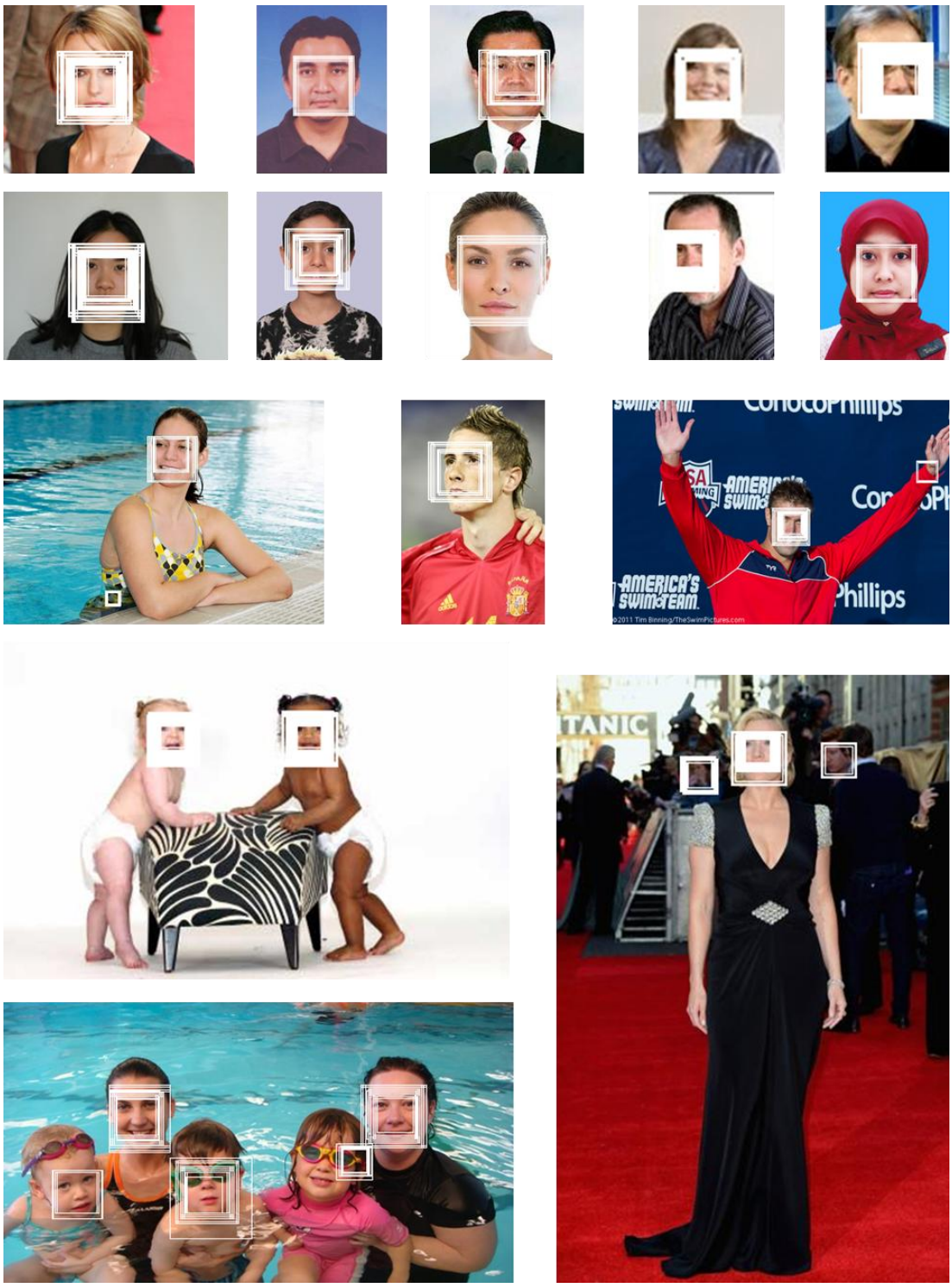
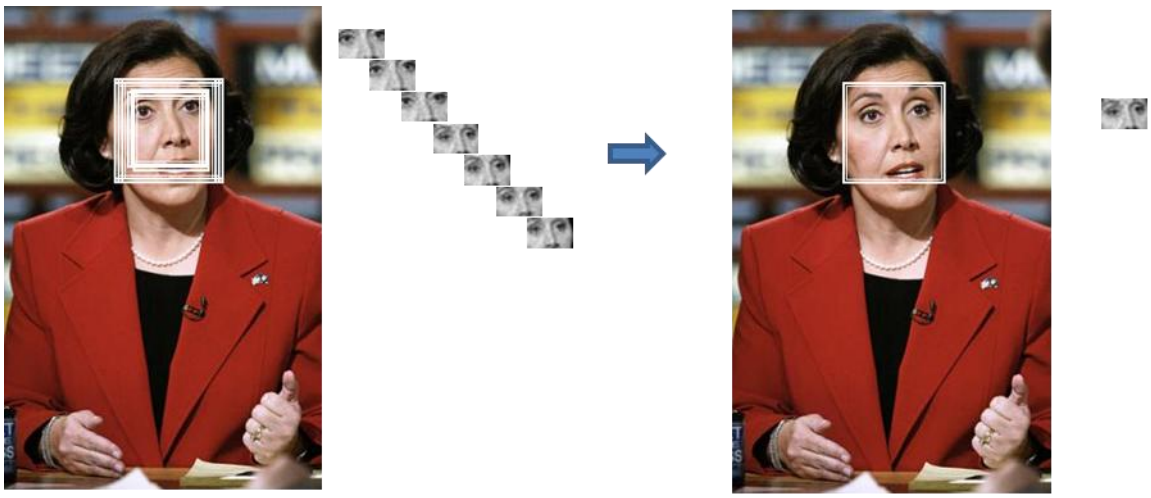


Figure 7.20: Overlapped detections examples.



Multiple overlapped detections (a)

Eliminating overlapped detections (b)

Figure 7.21: Eliminating overlapped detections (a) multiple overlapped detections; (b) eliminating overlapped detections into only one.

7.8 Experimental Results

Generally, when proposing a system with composite steps, it is compelling to evaluate each step separately. The testing and evaluation of the skin detection step is presented in Chapter 4. The experimental results of face-center localization are presented in Chapter 6. The experimental results in this section show the effectiveness of the general performance of the system.

Nowadays, there are many face databases available online for researchers. Grayscale face databases are used by Osuna et al. (1997), Rowley et al. (1998) and Schneiderman & Kanade (2000b) while other methods used color images (Garcia & Tziritas, 1999; Hsieh, Fan, & C., 2002; Srisuk et al., 2001; Zaqout et al., 2004), but their testing data are not available (NA).

To evaluate the proposed system, sample test images are collected from four public databases mentioned in chapter four: FEI, CVL, FDDB, and FSKTM (see Section 4.2). The system was tested using 550 images that contain frontal faces, with rotation ±10, and no occlusion. Images that imply other variations such as scale, number of faces, location, and ethnicity are included. These test images are divided into two sets of images: Test set A consists of 350

images containing single faces. Set B consists of 200 images containing 814 faces of different size, illumination, position and complex background. When a “face” is detected in the source image, the system draws an appropriate bounding box at the corresponding face. The bounding box is drawn larger than the study pattern size to correspond to actual face size (i.e. for clarification).

The system in the present work was executed in Matlab 2010a platform on Intel Core i5 at 2.2 GHz, 4GB DDR3 memory, and system type 64-bit, Window 7. Due to the fact that the proposed system is composed of several main steps with different methodologies and based on different features, the processing time required to detect human faces in the input image depends mainly on the image complexity (e.g. colors, size, texture, etc.).

Figure 7.22, Figure 7.23, Figure 7.24 and Figure 7.25 respectively show some face-detection results using the FEI dataset, CVL dataset, FSKTM dataset and FDDB dataset. As there are single face images (uniform and complex background) the detection rate is very good, that is about 95.42%. Considering the unconstrained nature of FSKTM and FDDB images that contain numerous faces, the detection rate is slightly lower, but remains highly satisfactory, that is 82.93%.

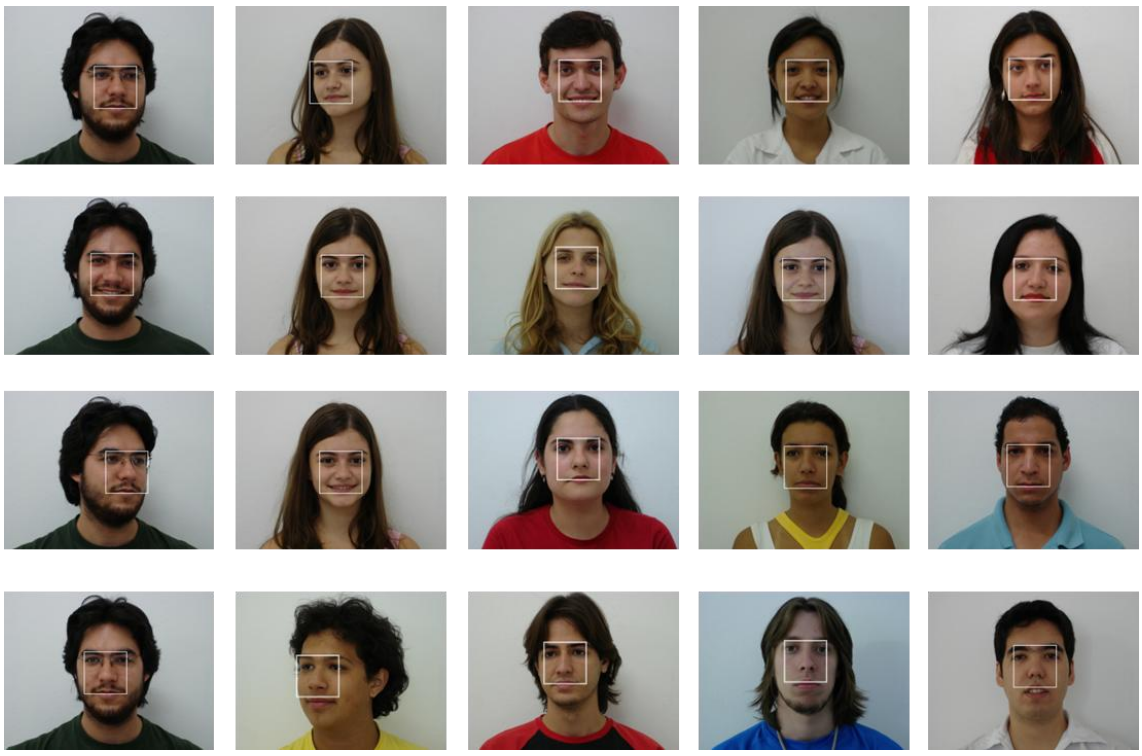


Figure 7.22: Some detection results using FEI dataset.

Figure 7.23: Some detection results using CVL dataset.

Figure 7.24: Some detection results using FSKTM dataset.

Figure 7.25: Some detection results using FDDB dataset.

Figure 7.25 (*continued*): Some detection results using FDDB dataset.

In general, false detections can hardly be avoided. Figure 7.26 shows examples of false detections. In this figure, the first row shows the whole face bounding box; while the second row shows the source pattern.

Figure 7.26: False detection examples; the first row shows the whole face box; while the second row shows the source pattern.

7.9 Comparison with Other Works

The comparative analysis of the proposed face detection system with other systems is shown in Table 7.5. The table shows the best results of different methods. Although different methods use different face databases in training and testing, the table gives an overall picture of the performance of the methods. The proposed system can detect between 82.93% and 95.42% of faces for a set of 550 images containing 1164 faces, revealing that the proposed system achieves detection and false rates which are equivalent to the best published results.

The remainder of this section presents a comparison with other works.

Table 7.5: Performance of the proposed face detector compared to other face detection methods

Method	Database used	Gray/Color	No. of faces	Detected Faces	Accuracy	False Detections
Skin color + LS Face Model (Hiremath & Danti, 2006)	CIT + FERET	Color	725	680	93.8%	NA
AdaBoost (Viola and Jones 2004)	MIT+CMU	Grayscale	507	130	76.1-94.1%	10-422
Skin color + PCA (Shih *et al.*, 2008)	NA	Color	486	470	96.7%	7.4%
Neural Networks (Rowley *et al.*, 1998)	CMU + FERET	Grayscale	507	390 - 461	86.2%	1/3,613,009
Bayes Classifier	FERET+NIST+CMU+HARVARD	Grayscale	441	124	85.5%	20.63%
Color Clustering (Hsieh, Fan, & C., 2002)	NA	Color	118	83	70.3%	6.6%
Fuzzy HSCC + SVM (Srisuk *et al.*, 2001)	NA	Color	NA	NA	93.6%	16.3%
Skin color + Wavelet Packet(Garcia & Tziritas, 1999)	NA	Color	104	98	94.23%	19.23%
Skin color + Ellipse fitting (Zaqout *et al.*, 2004)	NA	Color	NA	NA	~94.17%	~17.31%
HMM (Nefian, 1999)	MIT	Gray	432	NA	91.2 – 91.5%	56 - 43 (No.)
Support Vector Machine (Osuna *et al.*, 1997)	NA	Gray	155 + 313	115 + 304	74.2 - 97.1%	4 – 20 (No.)
Support Vector Machine (Kukenys, 2010)	CBCL+CMU+ UMIST+ Stirling	Gray	NA	NA	86.5%	105 (No.)
Proposed System (Set A & B)	CVL+ FEI+ FDDB+FSKTM	Color	350 + 814	334 + 675	82.93 - 95.42%	42 – 109 (No.)

Zaqout *et al.*(2004) have developed a two-phase face detection system in colored images. First, it applies skin color segmentation to detect skin-like regions based on lookup-tables in RGB color space. In the second stage, the system detects faces based on the assumption that the appearance of face is blob-like and has an approximately elliptical shape. The main differences are as follows:

- The skin detection approach of Zaqout's system used RGB color space. Despite its importance in image acquisition and display, the RGB color space is of limited use when processing color images, because it is not a perceptual model (Efford, 2000). This makes skin detection unreliable because of the high correlation among R, G, and B components. The R, G, and B components contain information related to color and intensity. However, the HSV color space is used in the current work for image segmentation. The HSV model is an ideal tool for developing image segmentation algorithms based on color descriptions (Chaves-González *et al.*, 2010; Efford, 2000; Gonzalez *et al.*, 2007). This makes the segmentation results of the approach taken in the current study more reliable.

- Zaqout's system uses uni-skin model, which makes it incapable of detecting changes in skin tones such as skin of people from different ethnic groups and illumination variations. If the skin-color model is too general, it may yield a large number of FP errors. On the other hand, if the skin-model is tight, then it may yield numerous FN errors. In contrast, a key contribution of this research work is the shift to multi-skin color models. Therefore, pixels that are indistinguishable in one model may be fully distinguishable in the other model that will suppress the false negatives. The system developed in the this work is applicable to images in more general situations since it is capable of detecting and separating skin color regions under various imaging conditions and among different ethnic groups. Furthermore, this allows the researchers to exploit more information about skin regions and the relationship between regions.

- Zaqout's system is based on the assumption that skin segmentation produces skin maps with approximately elliptical shape. Accordingly, an ellipse-fitting algorithm is used to detect human faces. In general, ellipse-fitting algorithm can be successfully used to detect human face from images under constrained imaging conditions such as canonical passport-like images. In the case of complex images that contain surfaces and objects with skin-like colors or other exposed parts of the human body such as hands, shoulders, legs, etc. the ellipse-fitting algorithm will fail and consequently show high false detections. However, in the current research work, the system does not depend on preconditions and assumptions concerning the detected skin regions but it resolves false alarms caused by skin-like regions through facial features extraction, rule-based geometrical knowledge, and neural network-based face detector. This makes our system more reliable and robust.

Concerning the classifier design, the classifier used in this work was originally inspired by the well-known pioneering face detection approach developed by Rowley *et al.* (1998), but has the following valuable improvements:

- Rowley's approach was originally designed to detect faces from gray scale images. The approach developed in the current study is to be used for color images.

- The search space of Rowley's approach was relatively high like other appearance-based face detectors. The classifier examines every location in the input image. Although, the sliding-window technique was widely used but this was very time consuming. However, in this present research two fast pre-processing classifiers are used: skin-color detector and face-center localization. These initial classifiers aim at rejecting the majority of pixels in the source image. Accordingly, they can restrict the search region of the ANNFD which makes the system faster.

- Restricting the search region of the ANNFD is also an important step to avoid false detections. As mentioned before, there are many natural non-face objects/patterns in the

real world, which look like face patterns when considered in separation. Excluding the background is an essential step to avoid such natural patterns.

- Rowley's approach used whole face pattern for training and operation. The presence or absence of mustache and beard along with mouth expressions makes training more difficult and consequently causes low detection rate. To improve the detection rate, the current study developed a system that uses partial face pattern instead of the whole face. The goal of the present study is to reduce face variability (for training and operation), and it is found (i.e. quantitatively and qualitatively) that using partial face pattern is better than using whole face pattern.

- Given that the ANN input is to be some representation of the image, one key design question is how to encode this image. Rowley's approach used pixel intensities of the face image as descriptors to train the neural network. In the current study, an additional 9 texture descriptors are computed for local regions in the face image, and then attached as input data with each image to train the neural network. This will enhance the reliability of the system, but at the expense of more computations. When the detection methods are used within systems, it is important to consider requirements, speed and accuracy. Accuracy may need to be sacrificed for speed or vice versa depending on the application.

- The system of the present study uses only one point, which is the "face center", to align all training faces automatically. This keeps the training faces as they are, whereas Rowley's approach used many points, which are used to normalize each training face to the same scale, orientation, and position. The transformations that are required for normalization may cause distortion to some faces due to the changes caused by the geometric transformations, since faces in their nature differ from one to another.

- Rowley's approach performed lighting correction process for each sliding window at classification run time. As reported by the authors, a dataset of 130 images required the system to process a total of 83,099,211 windows. This means that the lighting correction process was repeated millions of times and thereby the computational cost was high. In the current research work, the lighting correction process is done only once prior to the classification stage and the newly-generated lighting-corrected image is saved along with the source image. Therefore, there is no need to repeat this process at the operation phase of the classifier.

- Rowley *et al.* trained two neural networks. One network was moderately complex, focused on a small region of the image, and detected faces with a low false positive rate. They also trained a second neural network which was much faster, focused on larger regions of the image, and detected faces with a higher false positive rate. Rowley *et al.* used the faster second network to pre-screen the image in order to find candidate regions for the slower more accurate network. Our system extends this stage to include the X-Y-Reliefs classifier which shows effectiveness to reject a lot of non-face subimage windows early.

7.10 Discussion

In this chapter, an efficient face detector (or classifier) called ANNFD is presented. It is an appearance-based classifier that is capable of classifying a sub-image window as a face or non-face with high detection rate.

In this research, the ANNFD is initialized after two cascade fast classifiers; these are skin detector and face-center localization. This makes the search space of the ANNFD to be restricted to small hot spots of the image. Compared to the existing traditional classifiers, the ANNFD has many improvements such as using a novel partial face pattern, texture descriptors, and employing a semi-automatic method for preparing training faces. The partial face pattern is used efficiently to reduce face variability and consequently minimize the misrecognitions (i.e.,

FNs) caused by moustache, beard, as well as facial expressions. Texture descriptors are used to minimize the FPs. Training is done with minimum customization. The conclusion and direction of the future work are presented in the next chapter.

CHAPTER EIGHT

CONCLUSIONS AND IMPLICATION OF FUTURE DIRECTIONS

8.1 Research Findings and Achievements

By considering the diversity of image types, contents, and the range of objects that can be recognized in complex images, automatic face detection is one of the few attempts in computer vision for the recognition of object classes that admits a great deal of variability (non-rigid object). Hence, face detection researches have to deal with many challenges found in general purpose object detection. The inadequacy of automated face detectors is especially apparent when compared to our own innate face detection ability. We perform face detection, an extremely complex visual task, almost instantaneously and our own recognition ability is far more robust than any computer can hope to be.

The primary goal of this work is to develop a system that is capable to detect human faces from complex image(s). In this research we designed, developed, and implemented an efficient hybrid system to detect frontal faces independent of size, position, illumination, race, number of faces, and complex background. The general architecture of the system encompasses different methodologies to cope with the various challenges.

By considering the primary goal of this research, different goals with specific objectives are formulated. The following discussions show how these objectives are achieved.

- **Skin Color Modeling and Segmentation:** This objective has been achieved by developing a novel approach for human skin color modeling and segmentation using the HSV color space. To the best of our knowledge, the approach is the first attempt that shifts from the uni-skin model to the multi-skin models (Section 4.6). The approach does not require any initial parameters. Detecting skin regions is done without calculations because it is based on lookup table that makes it very fast (Section 4.11.2). It works with complex color

images, single and/or multi-faces, different ethnic groups, indoor or outdoor imaging conditions, and it is robust to background changes. The quantitative and qualitative results using four public datasets demonstrated the effectiveness and robustness of the proposed approach (Section 4.12). The first objective concerning skin color modeling and detection method of human targets in complex images is met.

In answering research question No. 4 concerning the applicability of this approach for other applications, the proposed skin detection approach can be used in many other applications such as hand gesture recognition, naked images filtering, teleconferencing, and other face processing applications (refer also to Section 4.13).

- **Testing and Evaluating Image Segmentation Methods:** Toward the achievement of the second objective of this research, we have presented a novel method for testing and evaluating the quality of skin segmentation algorithms and provided step-by-step procedure to use and compare different methods (see Section 4.10). The key contribution of this approach is based on creating a 2D standard set of test images for a legitimate and accurate evaluation. These images were generated automatically using the HSV color space. Additionally, detailed examples of such evaluation are presented. It is envisaged that this method reduces the cost and time required by developers to arrive at a practical system implementation. Research question No. 5 is dealt with as regards to the important characteristic of the image segmentation on its testing and evaluation in determining the performance.

- **Illumination Enhancement:** Due to the difficulty in controlling the lighting conditions in practical applications, developing a new method for automatic illumination enhancement is motivated to be the third objective of this thesis. It becomes an important preprocessing step to improve the performance of any face detector. In this research, a novel illumination enhancement approach using the idea of multi-layers image segmentations is presented (Section 5.2). The innovative approach for illumination correction is fast, simplified, reliable, and free of tuning parameters. The approach is based on local enhancement of skin

color tone rather than the entire image (research question No. 7 is answered). It has been shown that this approach is faster than other works because it is done only once prior to the classification stage (Section 5.4).

- **Rule-Based Geometrical knowledge:** The fourth objective of this research is to build a rule-based geometrical knowledge for face-center localization. The system in this stage is flexible and can be applied to the output of any skin detector and then return the position and extent of the candidate face-center region (Section 6.4). The advantage of using the rule–based approach is that the geometric relationships between the facial features are more invariant to changes in scaling, rotation, and face pose.

- **ANN-Based Face Detector (or Classifier)**: The fifth objective of this thesis is to develop an efficient appearance-based face detector based on machine learning techniques. In this research, a multi-layer neural network is used to build an appearance-based face detector. It is well known that even with the choice of a particular machine learning technique, the problem of face detection implies a number of sub-problems that need suitable solutions in order to achieve acceptable performance. The new solutions imply the following. ***1)*** A novel face model has been proposed, i.e. partial face pattern instead of whole face pattern (Section 7.4.1). To the best of our knowledge, this is the first attempt that uses partial face model. The qualitative and quantitative experiments show that this model has less variation than the traditional whole face model. ***2)*** A new semi-automatic method is proposed to prepare training faces instead of manual preparation (Section 7.4.3). ***3)*** A set of texture descriptors are proposed to enhance the feature vector. These descriptors are based on a collection of statistical properties and several pair-wise ordinal contrast relationships across facial regions (Section 7.5.2). By using the ratios of darkness between selected facial regions, the method becomes relatively insensitive to the illumination. This will enhance the reliability of the system (research question No. 10 is dealt with).

- **Conducting experiments for evaluating the proposed system:** The last objective is met by conducting different kinds of experiments for testing and evaluating each step of the proposed system separately and under different conditions. The experimental results and performance evaluation of each method are presented at the end of each relevant chapter.

The system can be used in many commercial and law enforcement applications such as face recognition systems, content-based indexing retrieval systems, banking and financial transactions, robotics, human computer interface, expression estimation, communications, facial expression recognition, and gender recognition. The general architecture of the system in this research balances the two often contradictory factors (speed and accuracy) whereby both factors are cautiously considered at various stages.

8.2 Conclusions

The problems associated with face detection especially the processing of complex images can rarely be solved through a single classifier (Sonka *et al.*, 2008). By considering this, it is common to combine a number of independent classifiers to improve the overall performance of the system. Often, these individual classifiers may be inadequate in isolation. Thus, a set of cascade classifiers based on diverse features such as skin-color, texture, geometric model, edges, and point features are considered in this research. Incorporation of all these features leads to a faster and more accurate system (research question No. 1 is answered).

This research divides the principal problem into several manageable sub-problems. Based on that, a hybrid face detection system which consists of several classification stages is developed. The robustness of the proposed hybrid architecture considers the integration of different methods and classifiers to achieve the ultimate goal.

In this section the conclusions are divided into three parts: Skin detection, face-center localization, and ANN-based face detector.

In relation to skin detection problem and based on the experimental results the following conclusions are drawn:

- Most colors in the colors set are non-skin colors which form about 90.36% of HSV color space (i.e. skin colors form about 9.64%). This amount of reduction in the color set is highly important to reject a majority of the image's pixels using fast pixel-based classifier. The subsequent complex classifiers will focus only on promising regions.

- Although excluding the background is an important step to speed up the system, it is also important to avoid false detections (i.e. natural non-face patterns that look like a face in isolation). This will increase the reliability of the system. The above-mentioned two points cover the answer to research question No. 2.

- In this research the limitation of uni-skin model is discussed. Although the uni-skin model is proposed by many previous works, it is inadequate to cover different skin color tones, such as dark shadow regions, blackish skin, strong light, makeup, and image reproduction. When the skin-color clustering model is too general, it may yield a large number of FPs. On the other hand, if the skin-model is tight, it may yield numerous FNs. The previous researchers had collected the training data (i.e. skin samples) and tried to build a skin color model for this purpose. They formulated the problem as a two-class classification problem that is skin and non-skin. Accordingly, all data of skin samples are treated as one model. A key contribution of our work is the analysis, interpretation, and classification of the collected data into different clusters based on skin color tone (i.e. different ethnic origins). If the objective is to locate faces of a particular race in an image (e.g. African), one should use skin samples from only that race. In this work, the skin samples for each ethnic group are collected separately and treated as different classes.

- This study disagrees with many previous works that argued and assumed there is no relationship between the chrominance components and luminance component. Consequently, they put the luminance component in the non-useful zone (Section 4.4.2). The present study showed that skin color differs in both intensity and chrominance. The study also showed that the relationship between intensity *V* and saturation *S* components is "Inverse nonlinear relationship" in HSV color space (Section 4.6).

- In answering research question No. 3 concerning the suitable color space for skin detection, in this study it is found that the HSV color space is a perfect tool for image segmentation due to many useful interesting properties (Section 4.3). Experimental results obtained using this color space were presented and analyzed (Section 4.10).

- Many random factors should be taken into account when selecting skin-color model such as the challenges mentioned in Section 3.4. In this study it is found that using multi-skin models can highly improve the detection rate based on a simple heuristic: since each skin model describes a specific skin color characteristics, pixels which are undetected in one model can be fully recognized in the other model. Accordingly, the skin detector based on multi-models shows high detection rate and shows robustness against variations in many random factors (research question No. 6 is answered).

- The use of multi-skin models in HSV color space facilitates easier and faster illumination correction.

- Image enhancement results show that local enhancement can highly improve face appearance that will be reflected positively on the general performance of the system. On the other hand, whole image enhancement cannot be generalized because when the original image is irregularly illuminated, some details on the resulting image will become either too bright or too dark.

- Image segmentation and skin color correction (illumination correction) are so closely related that they should not be performed separately. When developing a system in which image segmentation is an integral component, the choice of skin color modeling method used can directly affect the subsequent steps as a whole.

- Proposing a standard set of test images is important to evaluate and compare the characteristics of different segmentation methodologies, and thus determines their performance.

In relation to face-center localization system the following conclusions are drawn:

- Rule-based geometrical knowledge is a powerful approach to remove false alarms caused by objects with skin-like colors. The advantage of using rule–based approach is that the geometric relationships between the facial features are insensitive to scale, orientation, and face pose.

- The system in this stage can be applied to the output of any skin detector and returns the position and extent of the candidate face-center regions.

- Two methods for extracting facial features are used:
    - Threshold-based approach, and
    - Edge-based approach.

  Experimental results show that the edge-based approach shows more detection capabilities, although threshold-based is faster and more intuitive than edge-based approach.

  From the discussion above research question No. 8 is answered.

- Small faces may produce features extraction errors and consequently some geometric rules did not hold. In this research, small regions are passed to the subsequent stages leaving the final arbitration (i.e. face or none-face) to be done by the ANN-based classifier.

Consequently, the probability of missing small faces due to features extraction errors will be reduced.

In relation to ANN-based classifier and based on the experimental results the following conclusions are drawn:

- Training the classifier needs a lot of face and non-face training images. Although ANN-based classifier using pixel-based features can be trained and tested to show perfect performance on the training dataset, it would lead to poor performance when applied on real images. Thus, if we do not augment the input vector with adequate features, even the most sophisticated classifiers may fail to accomplish the classification task. The experimental results show that texture features highly improve the performance of the classifier (see Section 7.5).

- By combining the image and its texture descriptors, the new feature vector enhances the discriminating power of the classifier for face detection.

- Texture features extraction was performed using two approaches:
    - Statistical descriptors, and
    - DWT Daubechies-4 wavelet coefficients.

  In this research, it is found that texture features based on statistical descriptors show better classification results than wavelet coefficients (see Section 7.5.3).

- Ideally, we would want all training faces to be aligned with the least degree of variation so that the approximate positions of the facial features are the same. As we collect (or crop) the face images manually, this condition does not hold strictly. An efficient way to align all training faces automatically/semi-automatically is highly important. In this research, the proposed semi-automatic method improves the quality of the training data as well as reduces the time required for preparing training faces.

- In answering the research question No. 9 on using new face model for classification, the partial face model is proposed in this research. From the findings, it can be concluded that partial face model has the following properties: ***1)*** improves the quality of training data. ***2)*** makes it easier for the system to learn the function space of faces and consequently boosts the classifier's discrimination ability. ***3)*** has low-dimensionality.

- Although oval mask is widely used to reduce face variability and ensures that the system does not wrongly introduce any unwanted background structures into the face representation, it cannot cope with variations caused by moustache, beard, and mouth expressions.

The work presented in this thesis presents some new concepts and methods for resolving certain sub-problems implied in the main problem.

During these years of research, a large amount of data such as skin and non-skin pixels, face and non-face training images, face databases, and test images are collected and prepared that can be used by other researchers in the field. From the discussion above on the highlights of these research findings the ultimate aim of this work is achieved.

8.3 Implication of Future Directions

There are a number of directions for future work such as:

- Improving the general performance of the system including the detection rate and speed.
- Although skin detection and rule-based approaches are robust against rotation, but ANNFD was trained to detect frontal faces with ±10° rotated. It is desirable for the future work to extend the approach to detect faces of different poses, and faces rotated more than 10°.
- An important topic which has not been considered in this research is color enhancement of images containing strong colors that tend to be unreal (such as image that completely tends

to blue, red, etc.). It is an important pre-processing step that improves the general performance of any face processing system.

- Developing methods for detecting other parts of the human body (hair, shoulders, hands, etc.) can greatly improve the reliability of the system. It will increase the detection rate (TP) as well as reduce false positives (FP) caused by patterns which look like face pattern.
- The features in this work (color, texture, and facial) were selected specifically for face detection problem. In the future, it will be desirable to automatically learn which features are central so that the algorithm can be extended for general object detection problem.
- In face detection problem, the variability among faces in real complex images is so manifold that it is very difficult to build a unified model that can describe all these variations. The implementation of different methodologies shows cooperative effect between the system components. If the system is enhanced with new methodologies in future work, we may be able to achieve a much greater improvement in performance.
- Although we had been proposed an efficient semi-automatic method for face alignment, a fully automatic method is highly important to reduce the time and cost that are required for preparing training faces.
- The neural model is such that the neurons at each layer learn certain features from the input. It will be good to point to future analysis of the neural network model in terms of its structure.

REFERENCES

Adini, Y., Moses, Y., & Ullman, S. (1997). Face recognition: The problem of compensating for changes in illumination direction. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 19*(7), 721-732.

Adipranata, R., Ballangan, C.G., & Ongkodjojo, R.P. (2008). *Fast Method for Multiple Human Face Segmentation in Color Image.* Paper presented at the Second Int. Conf. on Future Generation Communication and Networking, 2, pp. 158-161, Hainan Island, China.

Agarwal, V., Abidi, B.R., Koschan, A., & Abidi, M.A. (2006). An overview of color constancy algorithms. *Journal of Pattern Recognition Research, 1*(1), 42-54.

Agathos, A., Pratikakis, I., Perantonis, S., Sapidis, N., & Azariadis, P. (2007). 3D mesh segmentation methodologies for CAD applications. *Computer-Aided Design and Applications, 4*(6), 827-841.

Aghbari, Z.A., & Al-Haj, R. (2006). Hill-manipulation: An effective algorithm for color image segmentation. *Image and Vision Computing, 24*(8), 894-903.

Agui, T., Kokubo, Y., Nagahashi, H., & Nagao, T. (1992). *Extraction of face regions from monochromatic photographs using neural networks.* Paper presented at the Proc. Second Int'l Conf. Automation, Robotics, and Computer Vision, 1, pp. 18.81-18.85.

Albiol, A., Torres, L., Bouman, C.A., & Delp, E. (2000). *A simple and efficient face detection algorithm for video database applications.* Paper presented at the International Conference on Image Processing, 2, pp. 239-242, Vancouver, BC, Canada.

An, G., Wu, J., & Ruan, Q. (2010). An illumination normalization model for face recognition under varied lighting conditions. *Pattern Recognition Letters, 31*(9), 1056-1067.

Apostol, T.M. (1979). Calculus II: Multi-Variable Calculus and Linear Algebra, with Applications to Differential Equations and Probability: John Wiley & Sons, New York.

Barnard, K., Finlayson, G., & Funt, B. (1997). Color constancy for scenes with varying illumination. *Computer Vision and Image Understanding, 65*(2), 311-321.

Baskan, S., Bulut, M.M., & Atalay, V. (2002). Projection based method for segmentation of human face and its evaluation. *Pattern Recognition Letters, 23*(14), 1623-1629.

Belaroussi, R., & Milgram, M. (2012). A comparative study on face detection and tracking algorithms. *Expert Systems with Applications, 39*(8), 7158-7164.

Bicego, M., Castellani, U., & Murino, V. (2003). *Using Hidden Markov Models and wavelets for face recognition.* Paper presented at the Proceedings 12th International Conference on Image Analysis and Processing.

Brand, J., & Mason, J.S. (2000). *A comparative assessment of three approaches to pixel-level human skin-detection.* Paper presented at the Proceedings. 15th International Conference on Pattern Recognition

Brown, D., Craw, I., & Lewthwaite, J. (2001). *A som based approach to skin detection with application in real time systems.* Paper presented at the Proc. of the British Machine Vision Conference.

Brunelli, R. (2009). *Template matching techniques in computer vision: Theor y and Practice*: John Wiley & Sons Ltd Publication.

Burdick, D., Calimlim, M., Flannick, J., Gehrke, J., & Yiu, T. (2005). Mafia: A maximal frequent itemset algorithm. *IEEE Transactions on Knowledge and Data Engineering, 17*(11), 1490-1504.

Burdick, H.E. (1997). *Digital Imaging: Theory and Applications*: McGraw-Hill, Inc.

Burges, C.J.C. (1998). A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery, 2*(2), 121-167.

Caetano, T.S., Olabarriaga, S.D., & Barone, D.A.C. (2002). *Performance evaluation of single and multiple-gaussian models for skin color modeling.* Paper presented at the Symposium on Computer Graphics and Image

Cai, J., & Goshtasby, A. (1999). Detecting human faces in color images. *Image and Vision Computing, 18*(1), 63-75.

Chai, D., & Ngan, K.N. (1999). Face segmentation using skin-color map in videophone applications. *Circuits and Systems for Video Technology, IEEE Transactions on, 9*(4), 551-564.

Chai, D., Phung, SL, & Bouzerdoum, A. (2003). A Bayesian skin/non-skin color classifier using non-parametric density estimation. *Proceedings of the International Symposium on Circuits and Systems, 2*, II-464-II-467.

Chang, Y.Z., Hung, K.T., & Lee, S.T. (2008). Human face detection with neural networks and the DIRECT algorithm. *Artificial Life and Robotics, 12*(1), 112-115.

Chaves-González, J. M., Vega-Rodríguez, M. A., Gómez-Pulido, J. A., & Sánchez-Pérez, J. M. (2010). Detecting skin in face recognition systems: A colour spaces study. *Digital Signal Processing, 20*(3), 806-823.

Chen, H.Y., Huang, C.L., & Fu, C.M. (2008). Hybrid-boost learning for multi-pose face detection and facial expression recognition. *Pattern Recognition, 41*(3), 1173-1185.

Chen, J.C., & Lien, J.J.J. (2009). A view-based statistical system for multi-view face detection and pose estimation. *Image and Vision Computing, 27*(9), 1252-1271.

Chen, Q., Wu, H., & Yachida, M. (1995). *Face detection by fuzzy pattern matching*. Paper presented at the Fifth International Conference on Computer Vision.

Chen , W. C., & Wang, M. S. (2007). Region-based and content adaptive skin detection in color images. *International journal of pattern recognition and artificial intelligence, 21*(5), 831.

Chen, W., Sun, T., Yang, X., & Wang, L. (2009). *Face detection based on half face-template.* Paper presented at the 9th International Conference on Electronic Measurement & Instruments.

Cheng, H.D., Jiang, XH, Sun, Y., & Wang, J. (2001). Color image segmentation: advances and prospects. *Pattern Recognition, 34*(12), 2259-2281.

Chin, TY. (2008). *Fuzzy skin detection.* (Thesis, Master of Science), Universiti Teknologi Malaysia.

Cho, K.M., Jang, J.H., & Hong, K.S. (2001). Adaptive skin-color filter. *Pattern Recognition, 34*(5), 1067-1073.

Clippingdale, S., & Fujii, M. (2011). *Skin Region Extraction and Person-Independent Deformable Face Templates for Fast Video Indexing*. Paper presented at the IEEE International Symposium on Multimedia (ISM), Dana Point CA, USA.

CMU Face Database. (2012). http://vasc.ri.cmu.edu/idb/html/face/

Colombo, A., C., Cusano, & R., Schettini. (2006). 3D face detection using curvature analysis. *Pattern Recognition, 39*(3), 444–455.

Craw, I., Ellis, H., & Lishman, J.R. (1987). Automatic extraction of face-features. *Pattern Recognition Letters, 5*(2), 183-187.

Curran, K., Li, X., & Mc Caughley, N. (2005). The use of neural networks in real-time face detection. *Journal of Computer Sciences, 1*(1), 47-62.

CVL Face Database. (2012). http://lrv.fri.uni-lj.si/facedb.html

Dai, Y., & Nakano, Y. (1996). Face-texture model based on SGLD and its application in face detection in a color scene. *Pattern Recognition, 29*(6), 1007-1017.

Dargham, J.A., & Chekima, A. (2006). Lips detection in the normalised RGB colour scheme. *Information and Communication Technologies ICTTA 1*, 1546-1551.

Daubechies , I. (1988). Orthonormal bases of compactly supported wavelets. *Communications on pure and applied mathematics, 41*(7), 909-996.

Daubechies, I. (1992). *Ten lectures on wavelets* (Vol. 61): CBMS-NSF Regional Conference Series In Applied Mathematics, SIAM.

Ding, L., & Martinez, A.M. (2010). Features versus context: An approach for precise and detailed detection and delineation of faces and facial features. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 32*(11), 2022-2038.

Ding, Z., Zhao, F., Shu, W., & Wu, M.Y. (2012). Face detection system for SVGA source with hecto-scale frame rate on FPGA board. *Microprocessors and Microsystems, 36*(4).

Do, H.C., You, J.Y., & Chien, S.I. (2007). Skin color detection through estimation and conversion of illuminant color under various illuminations. *Consumer Electronics, IEEE Transactions on, 53*(3), 1103-1108.

Doudpota, S. M., & Guha, S. (2012). *Automatic Actors Detection in Musicals*. Paper presented at the Seventh International Conference on Digital Information Management (ICDIM), pp. 226- 231, Macau, China.

Du, S., & Ward, R. (2005). *Wavelet-based illumination normalization for face recognition.* Paper presented at the IEEE International Conference on Image Processing.

Duan, L., Cui, G., Gao, W., & Zhang, H. (2002). *Adult image detection method base-on skin color model and support vector machine.* Paper presented at the The 5th Asian Conference on Computer Vision.

Duda, R.O., Hart, P.E., & Stork, D.G. (2001). *Pattern classification* (2nd ed.): John Willey & Sons.

Ebner, M. (2006). Evolving color constancy. *Special Issue on Evolutionary Computer Vision and Image Understanding of Pattern Recognition Letters, 27*(11), 1220-1229.

Ebner, M. (2007). *Color constancy*: John Wiley & Sons, England,.

Ebner, M., Tischler, G., & Albert, J. (2007). Integrating color constancy into JPEG2000. *Image Processing, IEEE Transactions on, 16*(11), 2697-2706.

Efford, N. (2000). *Digital Image Processing: A Practical Introduction Using Java* Addison-Wesley Longman Publishing Co., Inc.

Eveno, N., Caplier, A., & Coulon, P.Y. (2001). New color transformation for lips segmentation. *IEEE Fourth Workshop on Multimedia Signal Processing* 3-8.

Fasel, I., Fortenberry, B., & Movellan, J. (2005). A generative framework for real time object detection and classification. *Computer Vision and Image Understanding, 98*(1), 182-210.

FEI Face Database. (2012). http://fei.edu.br/~cet/facedatabase.html

FERET Face Database. (2012). http://www.nist.gov/itl/iad/ig/colorferet.cfm/

Fernandez, A., Ortega, M., Cancela, B., & Penedo, MG. (2012). Human Body Parts Contextual And Skin Color Region Information For Locating Human Body Parts. *Journal of Computer and information Technology 1* (1).

Fleuret, F., & Geman, D. (2001). Coarse-to-fine face detection. *International Journal of Computer Vision, 41*(1), 85-107.

Frisch, A. S., Vrschaeb, R., & Olanoc, A. (2007). Fuzzy fusion for skin detection. *Fuzzy Sets and Systems, 158*, 325-336.

Fu, K.S., & Mui, JK. (1981). A survey on image segmentation. *Pattern Recognition, 13*(1), 3-16.

Garcia, C., & Delakis, M. (2004). Convolutional face finder: A neural architecture for fast and robust face detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 26*(11), 1408-1423.

Garcia, C., & Tziritas, G. (1999). Face detection using quantized skin color regions merging and wavelet packet analysis. *Multimedia, IEEE Transactions on, 1*(3), 264-277.

Gasparini, F., & Schettini, R. (2006). *Skin segmentation using multiple thresholding.* Paper presented at the Proc. Internet Imaging VII

Gejguš, P., & Šperka, M. (2003). *Face tracking in color video sequences.* Paper presented at the Proceedings of the 19th spring conference on Computer graphics.

Ghazali, K.H.B., Ma, J., & Xiao, R. (2012). An Innovative Face Detection Based on YCgCr Color Space. *Physics Procedia, 25*, 2116-2124.

Ghiass, R.S., & Fatemizadeh, E. (2008). *Illumination and View Invariant Face Detection and Recognition in Images with Complex Background.* Paper presented at the Visual Media Production (CVMP 2008), 5th European Conference on Visual Media Production.

Gomez, G., Sanchez, M. , & Sucar, L.E. . (2002). On selecting colour components for skin detection. *16th International Conference on Pattern Recognition, 2*, 961-964 vol. 962.

Gomez, G., Sanchez, M., & Enrique Sucar, L. (2002). On selecting an appropriate colour space for skin detection. *MICAI 2002: Advances in Artificial Intelligence, Volume 2313*, 3-18.

Gonzalez, R.C., & Woods, R .E. (2002). *Digital Image Processing* (2 ed.): Prentice Hall Press.

Gonzalez, R.C., Woods, R.E., & Eddins, S.L. (2007). *Digital Image Processing Using MATLAB*: Prentice Hall Press.

Govindaraju, V. (1996). Locating human faces in photographs. *International Journal of Computer Vision, 19*(2), 129-146.

Greenspan, H., Goldberger, J., & Eshet, I. (2001). Mixture model for face-color modeling and segmentation. *Pattern Recognition Letters, 22*(14), 1525-1536.

Grossmann, A., & Morlet, J. (1984). Decomposition of Hardy functions into square integrable wavelets of constant shape. *SIAM journal on mathematical analysis, 15*(4), 723-736.

Guo, J.M., Lin, C.C., Wu, M.F., Chang, C.H., & Lee, H. (2011). Complexity Reduced Face Detection Using Probability-Based Face Mask Prefiltering and Pixel-Based Hierarchical-Feature Adaboosting. *IEEE Signal Processing Letters, 18*(8), 447-450.

Guo, J.M., & Wu, M.F. (2010). *Pixel-based hierarchical-feature face detection.* Paper presented at the Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on Acoustics Speech and Signal Processing.

Habili, N., Lim, C.C., & Moini, A. (2004). Segmentation of the face and hands in sign language video sequences using color and motion cues. *Circuits and Systems for Video Technology, IEEE Transactions on, 14*(8), 1086-1097.

Hadid, A., Pietikainen, M., & Ahonen, T. (2004). *A discriminative feature space for detecting and recognizing faces.* Paper presented at the Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.

Hagan, M.T., Demuth, H.B., & Beale, M.H. (1996). *Neural network design*: PWS Pub.

Hallinan, P.W. (1994). *A low-dimensional representation of human faces for arbitrary lighting conditions.* Paper presented at the Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.

Han, J., Awad, G., & Sutherland, A. (2009). Automatic skin segmentation and tracking in sign language recognition. *Computer Vision, IET, 3*(1), 24-35.

Haykin , S. (2005). *Neural Networks: A Comprehensive Foundation* (ninth indian reprint ed.): Prentice Hall

Haykin, S.S. (2009). *Neural networks and learning machines* (Vol. 3): Prentice Hall.

Heisele, B., Serre, T., Prentice, S., & Poggio, T. (2003). Hierarchical classification and feature reduction for fast face detection with support vector machines. *Pattern Recognition, 36*(9), 2007-2017.

Hiremath, PS, & Danti, A. (2006). Detection of multiple faces in an image using skin color information and lines-of-separability face model. *International Journal of Pattern Recognition and Artificial Intelligence, 20*(1), 39-62.

Hsieh, I-S., Fan, K-C., & C., Lin. (2002). A statistic approach to the detection of human faces in color nature scene. *Pattern Recognition, 35*(7), 1583-1596.

Hsu, R.L., Abdel-Mottaleb, M., & Jain, A.K. (2002). Face detection in color images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 24*(5), 696-706.

HU, B. (2011). *Skin Image Processing and Analysis Based on Pigment Separation it-Paper* Retrieved from http://www.it-paper.com/skin-image-processing-and-analysis-based-on-pigment-separation.html

Hua, LI, Abderrahim, E., Jaral, F., & Su, R. (2003). *An improved image segmentation approach based on level set and mathematical morphology.* Paper presented at the Third International Symposium on Multispectral Image Processing and Pattern Recognition.

Huang, J., Gutta, S., & Wechsler, H. (1996). *Detection of human faces using decision trees.* Paper presented at the Proceedings of the Second International Conference on Automatic Face and Gesture Recognition.

JAFFE Face Database. (2012). http://www.kasrl.org/jaffe.html

Jahne, B. (2004). *Practical handbook on image processing for scientific and technical applications* (2nd ed.): CRC Press.

Jahne, B. (2005). Digital Image Processing. (6th ed.): Springer, Berlin.

Jain, A.K. (1989). *Fundamentals of digital image processing*: Prentice-Hall, Inc.

Jebara, T., Russell, K., & Pentland, A. (1998). *Mixtures of eigenfeatures for real-time structure from texture*. Paper presented at the Sixth International Conference on Computer Vision.

Jeng, S.H., Liao, H.Y.M., Han, C.C., Chern, M.Y., & Liu, Y.T. (1998). Facial feature detection using geometrical face model: an efficient approach. *Pattern Recognition, 31*(3), 273-282.

Jin, H., Liu, Q., Lu, H., & Tong, X. (2004). *Face detection using improved LBP under bayesian framework.* Paper presented at the Third International Conference on Image and Graphics.

Jin , Z., Lou, Z., Yang, J., & Sun, Q. (2007). Face detection using template matching and skin-color information. *Neurocomputing, 70*(4-6), 794-800.

Johnson, K. (2012). Eigenfaces for recognition. from cmp.felk.cvut.cz/cmp/courses/recognition/Labs/pca/kimo.pdf

Jolliffe, I. (2005). *Principal component analysis*: Wiley Online Library.

Jones, M.J., & Rehg, J.M. (2002). Statistical color models with application to skin detection. *International Journal of Computer Vision 46(1),* 81–96.

Juang, C.F., & Shiu, S.J. (2008). Using self-organizing fuzzy network with support vector learning for face detection in color images. *Neurocomputing, 71*(16-18), 3409-3420.

Jun, B., & Kim, D. (2012). Robust face detection using local gradient patterns and evidence accumulation. *Pattern Recognition, 45*(9), 3304–3316.

Kakumanu, P., Makrogiannis, S., & Bourbakis, N. (2007). A survey of skin-color modeling and detection methods. *Pattern Recognition, 40*(3), 1106-1122.

Kanade, T. (1973). *Picture Processing by Computer Complex and Recognition of Human Faces* (PhD thesis), Kyoto Univ.

Kim, M., Park, J., & Joo, Y. (2005). New fuzzy skin model for face detection. *Advances in Artificial Intelligence, 3809,* 557-566.

Kinnebrock, W. (1995). *Neural Networks Fundamentals, Applicaltions, Examples* (2nd ed.): Galgotia Publications Pvt. Ltd.

Kirby, M., & Sirovich, L. (1990). Application of the Karhunen-Loeve procedure for the characterization of human faces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 12*(1), 103-108.

Kotropoulos, C., & Pitas, I. (1997). Rule-based face detection in frontal views. *IEEE International Conference on Acoustics, Speech, and Signal Processing, 4*, 2537-2540 vol. 2534.

Kovac, J., Peer, P., & Solina, F. (2003). Human skin colour clustering for face detection. *The IEEE Region 8 EUROCON 2003 Computer as a Tool 2*, 144-148

Krose, B., & Smagt, P. (1996). *An introduction to neural networks* (Eighth ed.): University of Amsterdam, The Netherlands.

Kukenys, I. (2010). *Human Face Detection with Support Vector Machines*. (Ph.D. Thesis), University of Otago.

Kumar, CNR, & Bindu, A. (2006). An efficient skin illumination compensation model for efficient face detection. *32nd Annual Conference on IEEE Industrial Electronics*, 3444-3449.

Kwon, Y.H., & da Vitoria Lobo, N. (1994). *Face detection using templates.* Paper presented at the Proceedings of the 12th International Conference on Pattern Recognition.

Lee, J. S., Kuo, Y. M., Chung, P. C., & Chen, E. L. (2007). Naked image detection based on adaptive and extensible skin color model. *Pattern Recognition, 40*(8), 2261-2270. doi: DOI 10.1016/j.patcog.2006.11.016

Lee, J.Y., & Yoo, S.I. (2002). *An elliptical boundary model for skin color detection.* Paper presented at the Proc. of the Int. Conf. on Imaging Science, Systems, and Technology.

Lee, M., & Park, C.H. (2008). *An efficient image normalization method for face recognition under varying illuminations.* Paper presented at the Proceedings of the 1st ACM international conference on Multimedia information retrieval.

Lei, L., Peng, J., & Yang, B. (2012). *72-trees index for image retrieval.* Paper presented at the IEEE 11th International Conference on Cognitive Informatics & Cognitive Computing, pp 268-273 Kyoto, Japan.

LFW Face Database. (2012). http://vis-www.cs.umass.edu/lfw/

Li, B., Xue, X., & Fan, J. (2007). A robust incremental learning framework for accurate skin region segmentation in color images. *Pattern Recognition, 40*(12), 3621-3632.

Li , S.Z., & Zhang, Z.Q. (2004). Floatboost learning and statistical face detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 26*(9), 1112-1123.

Li , Y. , & Xu, X. (2009). *Revolutionary Information System Application in Biometrics.* Paper presented at the Int. Conf. on Networking and Digital Society.

Lin, C. (2007). Face detection in complicated backgrounds and different illumination conditions by using YCbCr color space and neural network. *Pattern Recognition Letters, 28*(16), 2190-2200.

Lin, C., & Fan, K.C. (2001). Triangle-based approach to the detection of human face. *Pattern Recognition, 34*(6), 1271-1284.

Lin , S.H., Kung, S.Y., & Lin, L.J. (1997). Face recognition/detection by probabilistic decision-based neural network. *Neural Networks, IEEE Transactions on, 8*(1), 114-132.

Liu , C. (2003). A Bayesian discriminating features method for face detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 25*(6), 725-740.

Liu, X., & Cheng, T. (2003). *Video-based face recognition using adaptive hidden markov models.* Paper presented at the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Carnegie Mellon Univ., Pittsburgh, PA, USA.

Liu, Z., Yang, J., & Peng, N.S. (2005). An efficient face segmentation algorithm based on binary partition tree. *Signal Processing: Image Communication, 20*(4), 295-314.

Luger, G.F. (2005). *Artificial intelligence: Structures and strategies for complex problem solving*: Addison-Wesley Longman.

Ma, Z., & Leijon, A. (2010). *Human skin color detection in RGB space with Bayesian estimation of beta mixture models.* Paper presented at the 18th European Signal Processing Conference (EUSIPCO-2010), Aalborg, Denmark.

Mallat, S.G. (1987). *A compact multiresolution representation: the wavelet model*: Proc. IEEE Computer Society Workshop on Computer vision, IEEE Computer Society Press, Washington, D.C.

Mallat, S.G. (1989). A theory for multiresolution signal decomposition: the wavelet representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 11*(7), 674-693.

Marchand-Maillet, S., & Merialdo, B. (1999). *Pseudo two-dimensional hidden markov models for face detection in colour images.* Paper presented at the PROCEEDINGS OF THE AUDIO- AND VIDEO-BASED BIOMETRIC PERSON AUTHENTICATION (AVBPA'99).

Martinkauppi , B. (2002). *Face colour under varying illumination-analysis and application*. Ebook: http://herkules.oulu.fi/isbn9514267885/: Oulu University Library. .

Martinkauppi, B., Soriano, M., & Pietikainen, M. (2003). *Detection of skin color under changing illumination: a comparative study*. Paper presented at the Proceedings. 12th International Conference on Image Analysis and Processing, 2003. , Infotech Oulu, Oulu Univ., Finland

Martinkauppi, J.B., & Pietikäinen, M. (2005). Facial skin color modeling. In S. Z. Li & A. K. Jain (Eds.), *Handbook of face recognition* (pp. 113-135).

MATLAB. (2010). Version 7.11.0.584 (R2010b).

McKenna, S.J., Gong, S., & Raja, Y. (1998). Modelling facial colour and identity with gaussian mixtures. *Pattern Recognition, 31*(12), 1883-1892.

Menser, B., & Wien, M. (2000). *Segmentation and Tracking of Facial Regions iIn Color Image Sequences.* Paper presented at the Visual Communications and Image Processing 2000, Perth, Australia, 2000.

Miao, J., Yin, B., Wang, K., Shen, L., & Chen, X. (1999). A hierarchical multiscale and multiangle system for human face detection in a complex background using gravity-center template. *Pattern Recognition, 32*(7), 1237-1248.

Minsky, M., & Papert, S. . (1969). *Perceptrons*: Oxford, England, MIT Press.

MIT Face Database. (2012). http://cbcl.mit.edu/software-datasets/FaceData2.html

Mitchell, T.M. (1997). Machine learning. *Burr Ridge, IL: McGraw Hill*.

Moallem, P., Mousavi, B.S., & Monadjemi, S.A. (2011). A novel fuzzy rule base system for pose independent faces detection. *Applied Soft Computing, 11*(2011), 1801–1810.

Mohideen, S.K., Perumal, S.A., & Sathik, M.M. (2008). Image de-noising using discrete wavelet transform. *International Journal of Computer Science and Network Security, 8*(1), 213-216.

Morel, J.M., Petro, A.B., & Sbert, C. (2009). *Fast implementation of color constancy algorithms.* Paper presented at the Color Imaging XIV: Displaying, Processing, Hardcopy, and Applications, San Jose, CA, USA.

Munakata, T. (2008). *Fundamentals of the new artificial intelligence: neural, evolutionary, fuzzy and more* (2nd ed.): Springer-Verlag New York Inc.

Nair, P., & Cavallaro, A. (2009). 3-D face detection, landmark localization, and registration using a point distribution model. *Multimedia, IEEE Transactions on, 11*(4), 611-623.

Naji, S., Zainuddin, R., & Al-Jaafar, J. . (2010). *Automatic Illumination Correction for Human Skin*. Paper presented at the International Conference on Intelligent Network and Computing (ICINC 2010), Kuala Lumpur, Malaysia, 2010.

Naji, S., Zainuddin, R., & Jalab, H.A. (2012). Skin segmentation based on multi pixel color clustering models. *Digital Signal Processing*, *22* 933–940.

Naji, S., Zainuddin, R., Jallb, H.A., Zaid, M.A., & Eldouber, A. (2011). *Neural Network-Based Face Detection with Partial Face Pattern.* Paper presented at the The 2011 International Arab Conference on Information Technology ACIT 2011, Riyadh, Saudi Arabia.

Nefian, A. (1999). *A hidden Markov Model-based Approach for Face Detection and recognition.* (PhD. Thesis), Georgia Instiute of technology.

Nes, A. (2003). *Hybrid Systems for Face Recognition.* (Thesis Master of Science), Faculty of Information Technology, Norwegian University of Science and Technology.

Niese, R., Al-Hamadi, A., & Michaelis, B. (2007). A Novel Method for 3D Face Detection and Normalization. *Journal of Multimedia, 2*(5).

Ojala, T., Pietikäinen, M., & Harwood, D. (1996). A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition, 29*(1), 51-59.

Oliver, N., Pentland, A.P., & Berard, F. (2000). LAFTER: a real-time face and lips tracker with facial expression recognition. *Pattern Recognition, 33* 1369-1382.

Oskoei, Mohammadreza Asghari, & Hu, Huosheng. (2010). A survey on edge detection methods: Technical Report CES.

Osuna, E., Freund, R., & Girosi, F. (1997). *Training support vector machines: an application to face detection*. Paper presented at the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Juan, 1997.

Otsu, N. (1979). A Threshold Selection Method from Gray-Level Histograms,. *IEEE Transactions on Systems, Man, and Cybernetics, , 9*(1), 62-66.

Pal, S.K., & Mitra, S. (1999). *Neuro-fuzzy pattern recognition: methods in soft computing*: Wiley-Interscience.

Pandya, A.S., & Macy, R.B. (1996). *Pattern recognition with neural networks in C++*: CRC-Press.

Peer, P., & Solina, F. (1999). *An automatic human face detection method.* Paper presented at the Proceedings of Computer Vision Winter Workshop.

Pentland, A., Moghaddam, B., & Starner, T. (1994). *View-based and modular eigenspaces for face recognition.* Paper presented at the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Seattle, WA, 1994.

Phung, S.L., Bouzerdoum, A., & Chai, D. (2003). *Skin segmentation using color and edge information.* Paper presented at the Seventh International Symposium on Signal Processing and its Applications, 2003. , Paris, France.

Phung, S.L., Bouzerdoum, A., & Chai, D. (2005). Skin segmentation using color pixel classification: analysis and comparison. *IEEE transactions on pattern analysis and machine intelligence*, 148-154.

Phung, S.L., Chai, D., & Bouzerdoum, A. (2001). *A universal and robust human skin color model using neural networks.* Paper presented at the International Joint Conference on Neural Networks, IJCNN 2001. , Washington, DC.

Ping-Sing, T., & Shah, M. (1994). Shape from shading using linear approximation. *Image and Vision Computing, 12*(8), 487-498.

Pratt, W.K. (2001). *Digital image processing* (Third ed.): John Wiley & Sons, Inc.

Propp, M. , & Samal, A. (1992). Artificial Neural Network Architectures for Human Face Detection. *Intelligent Eng. Systems through Artificial Neural Networks, vol. 2*.

Provenzi, E., Fierro, M., Rizzi, A., De Carli, L., Gadia, D., & Marini, D. (2007). Random Spray Retinex: a new Retinex implementation to investigate the local properties of the model. *Image Processing, IEEE Transactions on, 16*(1), 162-171.

Raghuvanshi, D.S., & Agrawal, D. (2012). Human Face Detection by Using Skin Color Segmentation, Face Features and Regions Properties. *International Journal of Computer Applications, 38*(9).

Rajagopalan, AN, Kumar, K.S., Karlekar, J., Manivasakan, R., Patil, M.M., Desai, U.B., . . . Chaudhuri, S. (1998). *Finding faces in photographs.* Paper presented at the Sixth International Conference on Computer Vision, Bombay, India, 1998.

Rajasekaran, S., & Pai, G.A.V. (2004). *Neural networks, fuzzy logic and genetic algorithms: synthesis and applications*: PHI Learning Pvt. Ltd.

Rowley, H.A. (1999). *Neural network-based face detection.* (Ph.D Thesis), Carnegie Mellon University, Pittsburgh, PA, USA.

Rowley, H.A., Baluja, S., & Kanade, T. (1998). Neural network-based face detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 20*(1), 23-38.

Rowley, H.A., Jing, Y., & Baluja, S. (2006). *Large scale image-based adult-content filtering.* Paper presented at the International Conference on Computer Vision Theory and Applications Setúbal, Portugal.

Ruijsscher, B. (2006). *FPGA based accelerator for real-time skin segmentation.* Msc. thesis, Delft University of Technology, Mekelweg 4.

Ruiz-del-Solar, J., & Verschae, R. (2004). *Robust skin segmentation using neighborhood information.* Paper presented at the International Conference on Image Processing, , Singapore, 2004.

Russ, J.C. (2007). *The image processing handbook* (Fifth ed.): CRC Press, Taylor & Francis Group.

Russ, J.C., & Christian Russ, J. . (2008). *Introduction to image processing and analysis* CRC press.

Sakai, T., Nagao, M., & Fujibayashi, S. (1969). Line extraction and pattern detection in a photograph. *Pattern Recognition, 1*(3), 233-236, IN239-IN212, 237-248.

Salah, A., Bicego, M., Akarun, L., Grosso, E., & Tistarelli, M. (2007). *Hidden markov model-based face recognition using selective attention.* Paper presented at the Human Vision and Electronic Imaging XII, 649214, San Jose, CA, USA.

Samal, A., & Iyengar, P.A. (1995). Human face detection using silhouettes. *IJPRAI, 9*(6), 845-867.

Samaria, F., & Young, S. (1994). HMM-based architecture for face identification. *Image and Vision Computing, 12*(8), 537-543.

Sandeep, K., & Rajagopalan, AN. (2002). *Human face detection in cluttered color images using skin color and edge information.* Paper presented at the Indian Conference on Computer Vision, Graphics & Image Processing, ICVGIP 2002,Ahmadabad, India.

Sarfraz, M. S., Hellwich, O., & Riaz, Z. (2010). Feature Extraction and Representation for Face Recognition. In M. Oravec (Ed.): Intech open science.

Schneiderman, H., & Kanade, T. (1998). *Probabilistic modeling of local appearance and spatial relationships for object recognition.* Paper presented at the IEEE Computer Society Conference on computer Vision and Pattern Recognition, Santa Barbara, CA, USA. 1998.

Schneiderman, H., & Kanade, T. (2000). *A statistical method for 3D object detection applied to faces and cars*. Paper presented at the Proc. IEEE Conf. Computer Vision and Pattern Recognition, Hilton Head, SC, USA.

Sebe, N., Cohen, I., Huang, T.S., & Gevers, T. (2004). *Skin detection: A bayesian network approach.* Paper presented at the Proceedings of the 17th International Conference on Pattern Recognition, Cambridge, UK.

Seow, M.J., Valaparla, D., & Asari, V.K. (2003). *Neural network based skin color model for face detection.* Paper presented at the Proceedings of the 32nd Applied Imagery Pattern Recognition Workshop (AIPR'03), Washington, DC, USA.

Sevimli, H., Esen, E., Ateş, T.K., Ozan, E.C., Tekin, M., Loğoğlu, K.B., . . . Alatan, A.A. (2010). *Adult image content classification using global features and skin region detection.* Paper presented at the Proceedings of the 25th International Symposium on Computer and Information Sciences, London, UK.

Shan, S., Gao, W., Cao, B., & Zhao, D. (2003). *Illumination normalization for robust face recognition against varying lighting conditions.* Paper presented at the IEEE International Workshop on Analysis and Modeling of Faces and Gestures.

Shapiro, L. G., & Stockman, G. (2001). *Computer Vision* (1 ed.): Prentice Hall.

Shashua, A., & Riklin-Raviv, T. (2001). The quotient image: Class-based re-rendering and recognition with varying illuminations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 23*(2), 129-139.

Shih, F.Y. (2010). *Image Processing and Pattern Recognition: Fundamentals and Techniques*: john wiley & sons.

Shih, F.Y., Cheng, S., Chuang, C.F., & Wang, P.S.P. (2008). Extracting faces and facial features from color images. *International Journal of Pattern Recognition and Artificial Intelligence, 22*(3), 515-534.

Shin, M.C., Chang, K.I., & Tsap, L.V. (2002). *Does colorspace transformation make any difference on skin detection?* Paper presented at the Sixth IEEE Workshop on Applications of Computer Vision (WACV 2002), Orlando, Flordia, USA.

Sigal, L., Sclaroff, S., & Athitsos, V. (2000). *Estimation and prediction of evolving color distributions for skin segmentation under varying illumination.* Paper presented at the Proceedings IEEE Conference on Computer Vision and Pattern Recognition, 2000. , Hilton Head, SC, USA.

Sinha, P. (2002). Qualitative representations for recognition. *Biologically Motivated Computer Vision, 2525*, 129-146.

Sirohey, S.A. . (1993). Human Face Segmentation and Identification, *Technical Report CS-TR-3176 1993.*: Univ. of Maryland

Skarbek, W., Koschan, A., & Veroffentlichung, Z. (1994). Colour image segmentation-a survey: Tech. rep., Institute for Technical Informatics, Technical University of Berlin.

Smach, F., Atri, M., Mitéran, J., & Abid, M. (2006). Design of a neural networks classifier for face detection. *Journal of computer Science, 2*(3), 257-260.

Smith, L.I. (2002). A tutorial on principal components analysis. *Cornell University, USA.*

Sobottka, K., & Pitas, I. (1998). A novel method for automatic face segmentation, facial feature extraction and tracking. *Signal Processing: Image Communication, 12*(3), 263-281.

Solina, F., Peer, P., Batagelj, B., & Juvan, S. (2002). *15 seconds of fame-an interactive, computer-vision based art installation.* Paper presented at the 7th International Conference on Control, Automation, Robotics and Vision, ICARCV 2002., Singapore, 2002.

Sonka, M., Hlavac, V., & Boyle, R. (2008). *Image processing, analysis, and machine vision* (Third ed.): Thomson Corporation.

Soriano, M., Martinkauppi, B., Huovinen, S., & Laaksonen, M. (2000). *Skin detection in video under changing illumination conditions.* Paper presented at the 15th International Conference on Pattern Recognition, Barcelona, Spain.

Soriano, M., Martinkauppi, B., Huovinen, S., & Laaksonen, M. (2003). Adaptive skin color modeling using the skin locus for selecting training pixels. *Pattern Recognition, 36*(3), 681-690.

Soulie, F.F., Viennet , E., & Lamy, B. (1993). Multi-modular neural network architectures: applications in optical character and human face recognition. *International journal of pattern recognition and artificial intelligence, 7*(04), 721-755.

Srisuk, S., & Kurutach, W. (2002). *A new robust face detection in color images.* Paper presented at the IEEE Fifth International Conference on Automatic Face and Gesture Recognition. , Washington, D.C., USA.

Srisuk, S., Kurutach, W., & Limpitikeat, K. (2001). A Novel Approach For Robust, Fast And Accurate Face Detection. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 9*(6), 769-779.

Starck, J.L., Murtagh, F.D., & Bijaoui, A. (1998). *Image processing and data analysis: the multiscale approach*: Cambridge University Press.

Starovoitov, VV, Samal, DI, Briliuk, DV, & Kopendakov, A. (2003). *Image enhancement for face recognition.* Paper presented at the International Conference on Iconics 2003, Saint-Petersburg, Russia.

Stollnitz, E.J., DeRose, T.D., & Salesin, D.H. (1996). *Wavelets for computer graphics: theory and applications*: (The Morgan Kaufmann Series in Computer Graphics) Morgan Kaufmann Pub.

Storring, M. (2004). *Computer vision and human skin colour*. (Ph. D. thesis), Aalborg University, Denmark.

Su, M.C., & Chou, C.H. (1999). *Application of associative memory in human face detection.* Paper presented at the International Joint Conference on Neural Networks IJCNN 1999, Washington, D.C., USA.

Sung, K.K., & Poggio, T. (1998). Example-based learning for view-based human face detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 20*(1), 39-51.

Tan, W., Chan, C., Yogarajah, P., & Condell, J. (2012). A Fusion Approach for Efficient Human Skin Detection. *Industrial Informatics, IEEE Transactions on*(99), 1-1.

Taqa, A.Y., & Jalab, H. (2010). Increasing the reliability of skin detectors. *Scientific Research and Essays, 5*(17), 2480-2490.

Terrillon, J.C., David, M., & Akamatsu, S. (1998). *Detection of human faces in complex scene images by use of a skin color model and of invariant Fourier-Mellin moments*. Paper presented at the Fourteenth International Conference on Pattern Recognition, Brisbane, Qld., Australia.

Terrillon, J.C., Shirazi, M.N., Fukamachi, H., & Akamatsu, S. (2000). *Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images.* Paper presented at the Fourth IEEE International Conference on Automatic Face and Gesture Recognition, 2000. , Grenoble, France.

Tobin, D.J. (2006). Biochemistry of human skin—our brain on the outside. *Chem. Soc. Rev., 35*(1), 52-67.

Tomaz, F., Candeias, T., & Shahbazkia, H. (2004). *Fast and accurate skin segmentation in color images.* Paper presented at the First Canadian Conference on Computer and Robot Vision, London, UK, Ontario, Canada.

Tsao, W.K., Lee, A.J.T., Liu, Y.H., Chang, T.W., & Lin, H.H. (2010). A data mining approach to face detection. *Pattern Recognition, 43*(3), 1039-1049.

Tsekeridou, S., & Pitas, I. (1998). *Facial feature extraction in frontal views using biometric analogies.* Paper presented at the European Association for Signal Processing Eusipco - 98, Island of Rhodes, Greece.

Turk, M.A., & Pentland, A. (1991a). Eigenfaces for recognition. *Journal of cognitive neuroscience, 3*(1), 71-86.

Turk, M.A., & Pentland, A. (1991b). *Face recognition using eigenfaces*.

Unnikrishnan, R., Pantofaru, C., & Hebert, M. (2007). Toward objective evaluation of image segmentation algorithms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 29*(6), 929-944.

Vaclav, Matyas, Jr., & Zdenek, Riha. (2001). Biometric Authentication Systems. www.ecom-moitor.com

Vadakkepat, P., Lim, P., De Silva, L.C., Jing, L., & Ling, L.L. (2008). Multimodal approach to human-face detection and tracking. *Industrial Electronics, IEEE Transactions on, 55*(3), 1385-1393.

Vaillant, R., Monrocq, C., & Le Cun, Y. (1994). Original approach for the localisation of objects in images. *IEE Proceedings Vision, Image and Signal Processing, 141*(4), 245-250.

Verma, R.C., Schmid, C., & Mikolajczyk, K. (2003). Face detection and tracking in a video by propagating detection probabilities. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 25*(10), 1215-1228.

Vezhnevets, V., Sazonov, V., & Andreeva, A. (2003). A survey on pixel-based skin color detection techniques. *Graphicon 3*, 85-92.

Viola, P., & Jones, M. (2004). Robust real-time face detection. *International Journal of Computer Vision, 57*(2), 137-154.

Wang, H., Leng, Y., Wang, Z., & Wu, X. (2007). Application of image correction and bit-plane fusion in generalized PCA based face recognition. *Pattern Recognition Letters, 28*(16), 2352-2358.

Wang , H., Li, S.Z., & Wang, Y. (2004). *Face recognition under varying lighting conditions using self quotient image.* Paper presented at the Sixth IEEE International Conference on Automatic Face and Gesture Recognition, Seoul, Korea.

Wang, J., & Tan, T. (2000). A new face detection method based on shape information. *Pattern Recognition Letters, 21*(6-7), 463-471.

Wang, P., & Ji, Q. (2007). Multi-view face and eye detection using discriminant features. *Computer Vision and Image Understanding, 105*(2), 99-111.

Wang , Y., & Yuan, B. (2001). A novel approach for human face detection from color images under complex background. *Pattern Recognition, 34*(10), 1983-1992.

Wei, G., & Sethi, I.K. (2000). *Omni-face detection for video/image content description.* Paper presented at the the ACM workshops on Multimedia, Los Angeles, CA, USA.

Wu, B., Ai, H., Huang, C., & Lao, S. (2004). *Fast rotation invariant multi-view face detection based on real adaboost*. Paper presented at the Sixth IEEE International Conference on Automatic Face and Gesture Recognition, Seoul, Korea.

Wu , H., Chen, Q., & Yachida, M. (1999). Face detection from color images using a fuzzy pattern matching method. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 21*(6), 557-563.

Wu, J., Brubaker, S.C., Mullin, M.D., & Rehg, J.M. (2008). Fast asymmetric learning for cascade face detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 30*(3), 369-382.

Xiao, R., Li, M.J., & Zhang, H.J. (2004). Robust multipose face detection in images. *Circuits and Systems for Video Technology, IEEE Transactions on, 14*(1), 31-41.

Xiaohua, L., Lam, K.M., Lansun, S., & Jiliu, Z. (2009). Face detection using simplified Gabor features and hierarchical regions in a cascade of classifiers. *Pattern Recognition Letters, 30*(8), 717-728.

Xie, X., & Lam, K.M. (2005). Face recognition under varying illumination based on a 2D face shape model. *Pattern Recognition, 38*(2), 221-230.

XM2VTS Face Database. (2102). http://www.ee.surrey.ac.uk/CVSSP/xm2vtsdb/

Yang, C.H.T., Lai, S.H., & Chang, L.W. (2002). *Robust face matching under different lighting conditions*. Paper presented at the IEEE International Conference on Multimedia and Expo ICME 2002, Lausanne, Switzerland.

Yang, G., & Huang, T.S. (1994). Human face detection in a complex background. *Pattern Recognition, 27*(1), 53-63.

Yang , H.M., Kriegman, D. J., & Ahuja, N. (2002). Detecting faces in images: A survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence, 24*, 34-58.

Yang, M., Crenshaw, J., Augustine, B., Mareachen, R., & Wu, Y. (2010). AdaBoost-based face detection for embedded systems. *Computer Vision and Image Understanding, 114*(11), 1116-1125.

Yang, M.H. (2000). *Hand gesture recognition and face detection in images.* (Ph.D. Thesis), University of Illinois, Urbana, Illinois.

Yang, M.H., & Ahuja, N. (1998). *Detecting human faces in color images.* Paper presented at the IEEE International Conference on Image Processing (ICIP-98), Chicago, Illinois, USA. pp. 127-130.

Yang, M.H., Kriegman, D., & Ahuja, N. (2001). Face detection using multimodal density models. *Computer Vision and Image Understanding, 84*(2), 264-284.

Yang, M.H., Roth, D., & Ahuja, N. (2000). A SNoW-based face detector. In S. A. Solla, T. K. Leen & K. R. Müller (Eds.), *Advances in Neural Information Processing Systems* (Vol. 12, pp. 855-861): MIT Press.

Yow, K.C., & Cipolla, R. (1997). Feature-based human face detection. *Image and Vision Computing 15*(9), 713-735.

Yuetao, D., & Nana, Y. (2011). Research of Face Detection in Color Image Based on Skin Color. *Energy Procedia, 13*, 9395-9401.

Yuille, A.L., Hallinan, P.W., & Cohen, D.S. (1992). Feature extraction from faces using deformable templates. *International Journal of Computer Vision, 8*(2), 99-111.

Zaidan, AA, Ahmed, NN, Karim, H.A., Alam, G.M., & Zaidan, BB. (2010). Increase reliability for skin detector using backprobgation neural network and heuristic rules based on YCbCr. *scientific research and essays, 5*(19), 2931-2946.

Zainuddin , R., & Naji, S. . (2010). *Multi-skin color clustering models for face detection.* Paper presented at the Second International Conference on Digital Image Processing, Proceedings of SPIE, Singapore.

Zainuddin, R., Naji, S., & Al-Jaafar, J. (2010). Suppressing False Negatives in Skin Segmentation. In T. Kim, Y. Lee, B. Knag & D. Slezak (Eds.), *Lecture Notes in Computer Science LNCS* (Vol. 6485, pp. 136-144).

Zaqout, I. (2006). *An integrated approach for detecting human faces in color images.* (Ph.D. Thesis), University of Malaya.

Zaqout , I., Zainuddin, R., & Baba, S. (2004). Human Face Detection In Color Images. *Advances in Complex Systems, Vol. 7*(3), 369–383.

Zarit, B.D., Super, B.J., & Quek, F.K.H. (1999). *Comparison of five color models in skin pixel classification.* Paper presented at the International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems ICCV '99, pp 58-63, Corfu, Greece.

Zhang , H., Fritts, J.E., & Goldman, S.A. (2008). Image segmentation evaluation: A survey of unsupervised methods. *Computer Vision and Image Understanding, 110*(2), 260-280.

Zhang , L., Chu, R., Xiang, S., Liao, S., & Li, S. (2007). Face detection based on multi-block lbp representation. *Advances in Biometrics, Lecture Notes in Computer Science LNCS* (Vol. 4642, pp. 11-18).

Zhang, W., & Zelinsky, G. (2004). Current Advances in Computer-based Object Detection and Target Acquisition. *Technical Report EYECOG-04-01*: State University of New York at Stony Brook.

Zhao-yi, P., Yu, Z., & Ping, W. (2009). *Multi-pose face detection based on adaptive skin color and structure model.* Paper presented at the International Conference on Computational Intelligence and Security CIS '09. 1, pp. 325-329, Beijing, China.

Zhao, W.Y., & Chellappa, R. (2000). *Illumination-insensitive face recognition using symmetric shape-from-shading.* Paper presented at the IEEE Conference on Computer Vision and Pattern Recognition, 1, pp. 286-293, Hilton Head Island, SC, USA.

Zhu, J. (2005). *Pattern modeling and classification in vision systems.* (Ph.D. Thesis), Princeton University.

Zou, X., Kittler, J., & Messer, K. (2007b). *Illumination invariant face recognition: A survey.* Paper presented at the IEEE International Conference on Biometrics: Theory, Applications, and Systems. pp 1-8, Crystal City, Virginia, USA.

APPENDIX-A

List of Publications

Article in Academic Journals

Naji, S., Zainuddin R., Sameem A.K., Jalab H.A., *"Detecting Faces in Colored Images Using Multi-skin Color Models and Neural Network with Texture Analysis".* Malaysian Journal of Computer Science, Vol. 26, Issue 2 (2013), pp. 101-123. (*ISI-Cited Publication, Q4)*

Naji, S., Zainuddin, R., & Jalab, H.A., "*Skin segmentation based on multi pixel color clustering models".* Digital Signal Processing, Vol. 22, Iss. 6 (2012), pp. 933–940 *(ISI/SCOPUS Cited Publication, Q1)*

Chapter in Book

Zainuddin, R., Naji, S., & Al-Jaafar, J., "*Suppressing False Negatives in Skin Segmentation"*. In T. Kim, Y. Lee, B. Knag & D. Slezak (Eds.), Lecture Notes in Computer Science LNCS Vol. 6485 (2010), pp. 136-144. (*Springer Publication,)*

Proceeding

Naji, S., Zainuddin, R., Jallb, H.A., Zaid, M.A., & Eldouber, A., *"Neural Network-Based Face Detection with Partial Face Pattern".* Paper presented at the 2011 International Arab Conference on Information Technology ACIT 2011, Riyadh, Saudi Arabia, pp. 1-7.

Zainuddin , R., & Naji, S*., "Multi-skin color clustering models for face detection".* Paper presented at the Second International Conference on Digital Image Processing, Proceedings of SPIE 2011, Singapore, Vol. 7546, pp.1-10.

Naji, S., Zainuddin, R., & Al-Jaafar, J., *"Automatic Illumination Correction for Human Skin".* Paper presented at the International Conference on Intelligent Network and Computing ICINC 2010, Kuala Lumpur, Malaysia, Vol. 2, pp. 151-156

APPENDIX-B

Back-Propagation Learning Algorithm - (Sonka, 2008)

1. Assign small random numbers to the weights ω_{ij}, and set $k = 0$.
2. Input a pattern **v** from the training set and evaluate the neural net output **y**.
3. If **y** does not match the required output vector T, adjust the weights

$$\omega_{ij}(k+1) = \omega_{ij}(k) + \varepsilon\delta_j z_i(k) + \alpha(\omega_{ij}(k) - \omega_{ij}(k-1))$$

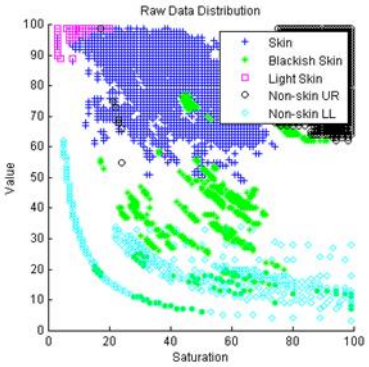
where ε is called the **learning rate**, α is called the **momentum constant**, $z_i(k)$ is the output of node i, k is the iteration number, δ_j is an error associated with the node j in the adjacent upper level

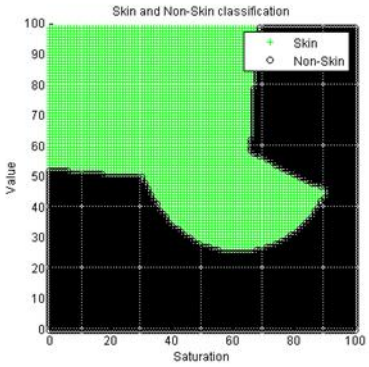
$$\delta_j = \begin{cases} y_j(1-y_j)(T_j - y_j) & \textit{for output node } j, \\ z_j(1-z_j)\sum_l \delta_l \omega_{jl} & \textit{for hidden node } j \end{cases}$$

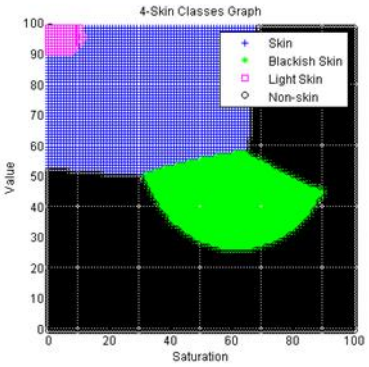
4. Go to step 2 and fetch the next input pattern.
5. Increment k, and repeat step 2 to 4 until each training pattern outputs a suitably good approximation to that expected. Each circuit of this loop is termed an **epoch**.

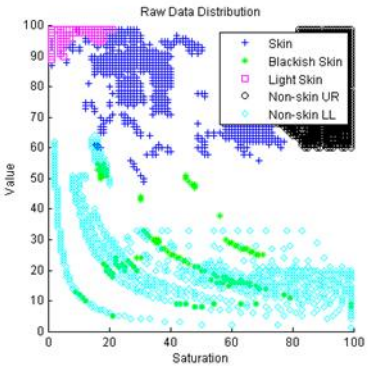
APPENDIX-C

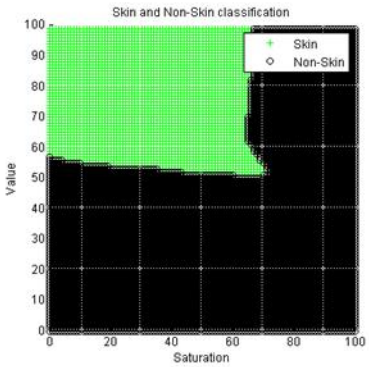
Skin detection results using Bayes classifier based on two-class classification problem

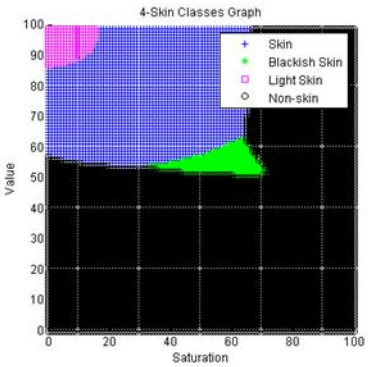


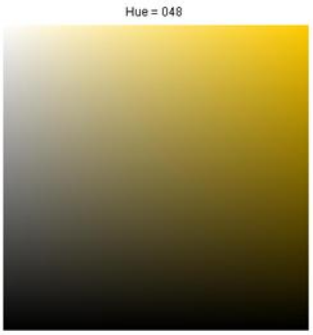


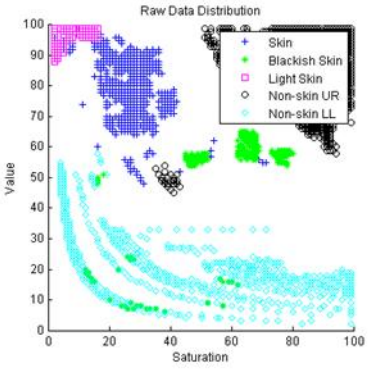


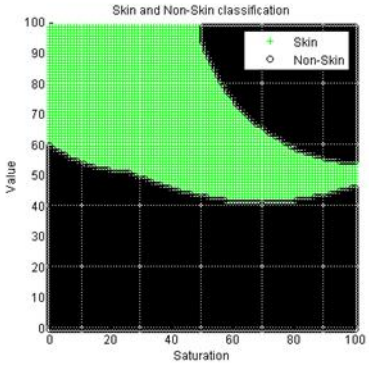


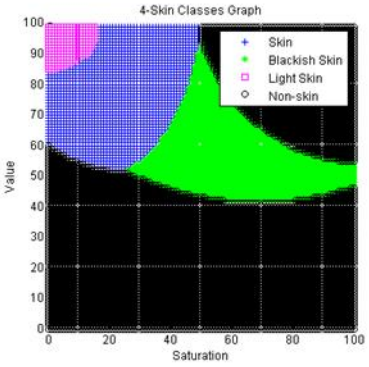




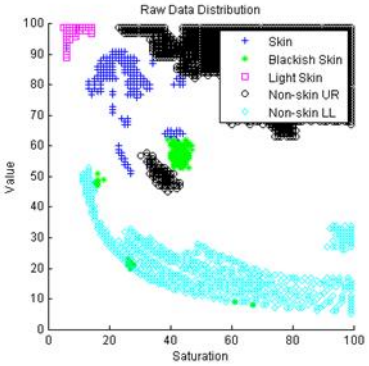

Hue = 048

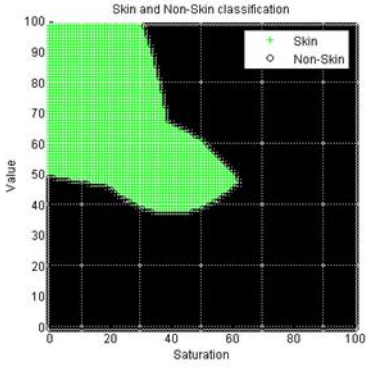

Raw Data Distribution
Skin
Blackish Skin
Light Skin
Non-skin UR
Non-skin LL
Value
Saturation

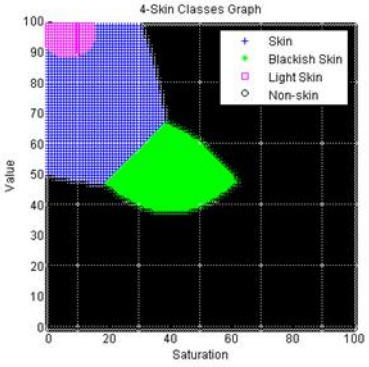

Skin and Non-Skin classification
Skin
Non-Skin
Value
Saturation


4-Skin Classes Graph
Skin
Blackish Skin
Light Skin
Non-skin
Value
Saturation

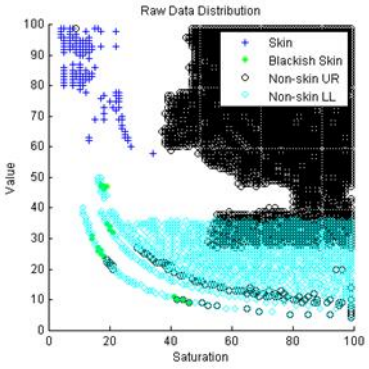

Hue = 054

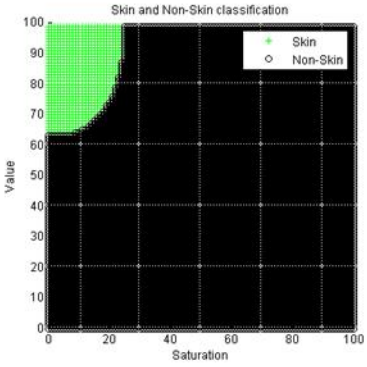

Raw Data Distribution
Skin
Blackish Skin
Light Skin
Non-skin UR
Non-skin LL
Value
Saturation

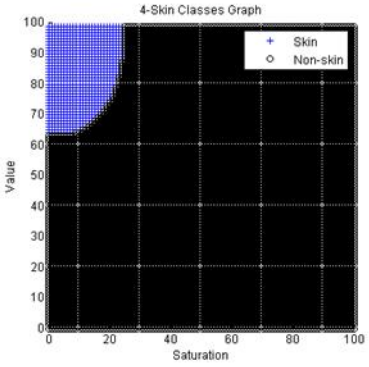

Skin and Non-Skin classification
Skin
Non-Skin
Value
Saturation


4-Skin Classes Graph
Skin
Blackish Skin
Light Skin
Non-skin
Value
Saturation

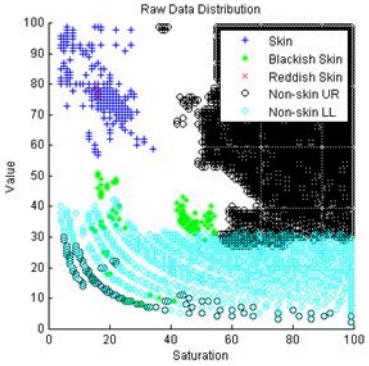

Hue = 336

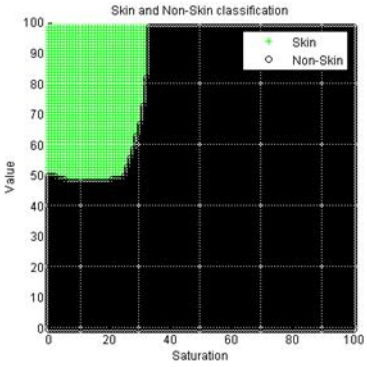

Raw Data Distribution
Skin
Blackish Skin
Non-skin UR
Non-skin LL
Value
Saturation

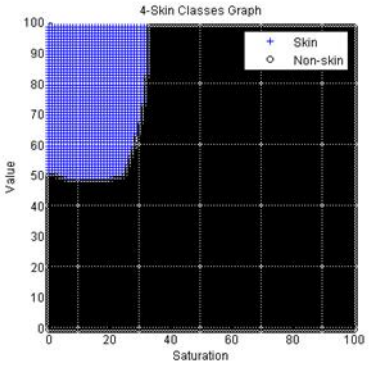

Skin and Non-Skin classification
Skin
Non-Skin
Value
Saturation


4-Skin Classes Graph
Skin
Non-skin
Value
Saturation


Hue = 342

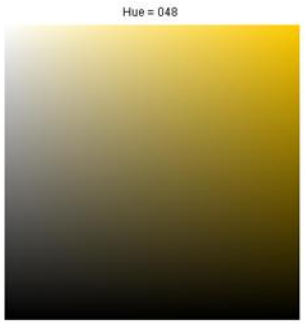

Raw Data Distribution
Skin
Blackish Skin
Reddish Skin
Non-skin UR
Non-skin LL
Value
Saturation

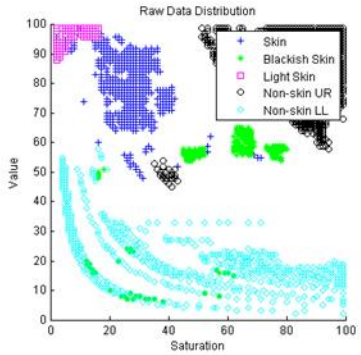

Skin and Non-Skin classification
Skin
Non-Skin
Value
Saturation

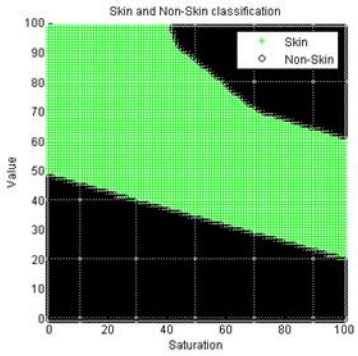

4-Skin Classes Graph
Skin
Non-skin
Value
Saturation

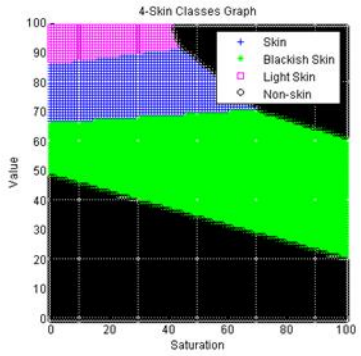
APPENDIX-D

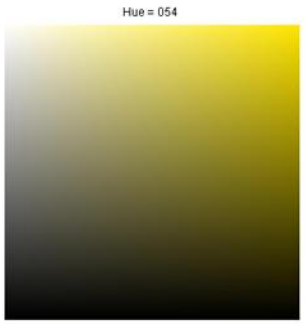
Skin detection results using LDA classifier – Continued

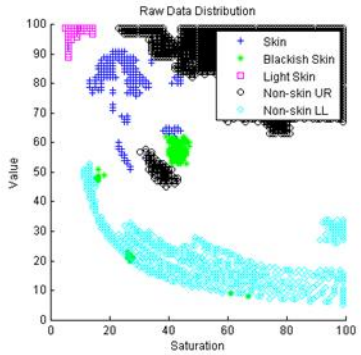


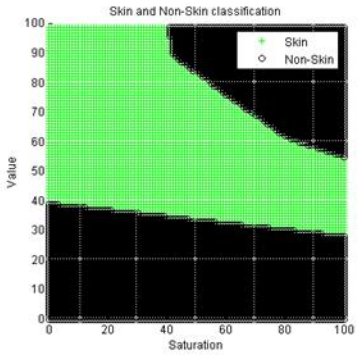


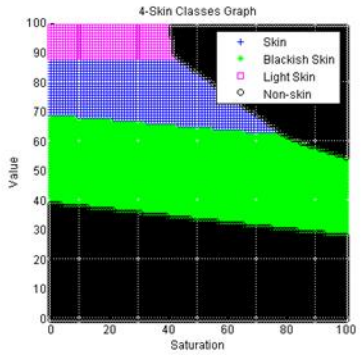




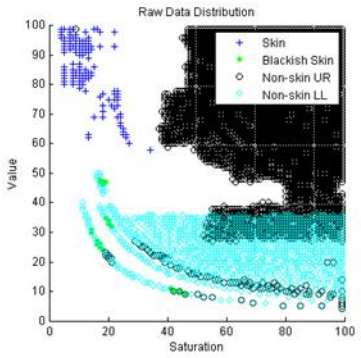


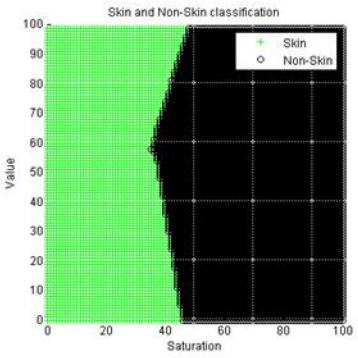


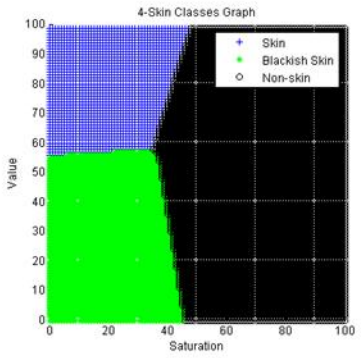




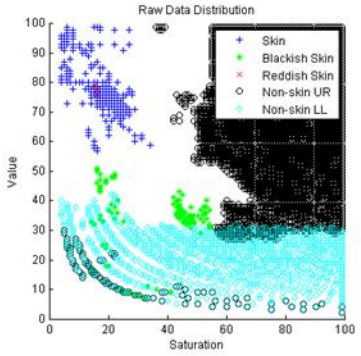

Hue = 336

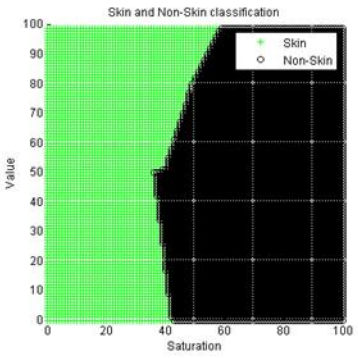

Raw Data Distribution
Skin
Blackish Skin
Non-skin UR
Non-skin LL
Value
Saturation

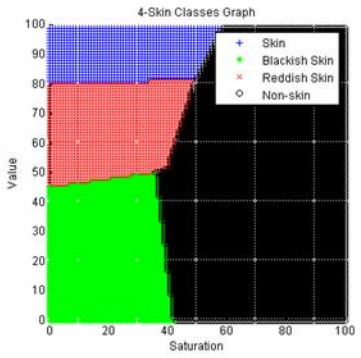

Skin and Non-Skin classification
Skin
Non-Skin
Value
Saturation


4-Skin Classes Graph
Skin
Blackish Skin
Non-skin
Value
Saturation

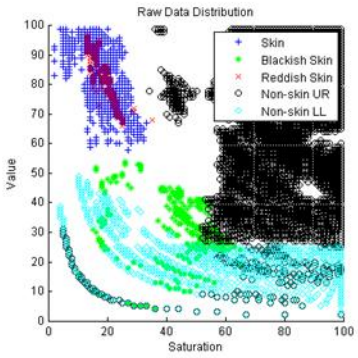

Hue = 342

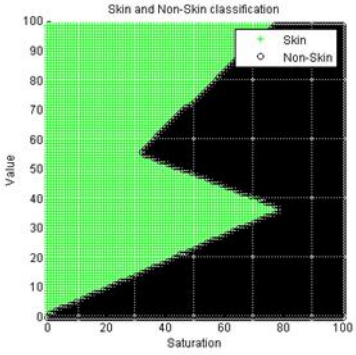

Raw Data Distribution
Skin
Blackish Skin
Reddish Skin
Non-skin UR
Non-skin LL
Value
Saturation

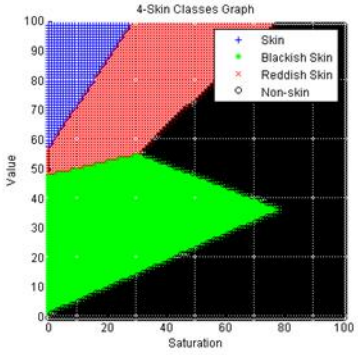

Skin and Non-Skin classification
Skin
Non-Skin
Value
Saturation


4-Skin Classes Graph
Skin
Blackish Skin
Reddish Skin
Non-skin
Value
Saturation

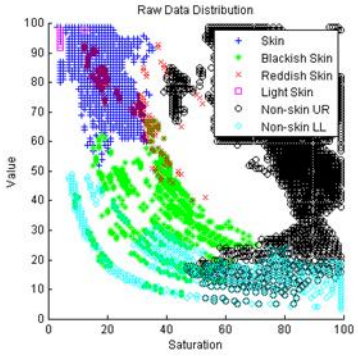

Hue = 348

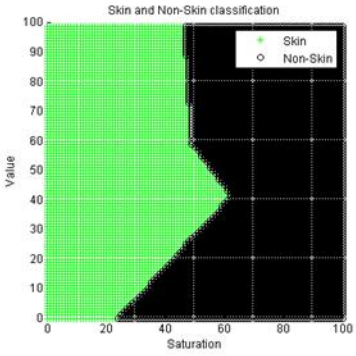

Raw Data Distribution
Skin
Blackish Skin
Reddish Skin
Non-skin UR
Non-skin LL
Value
Saturation

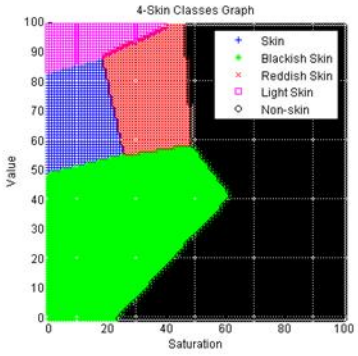

Skin and Non-Skin classification
Skin
Non-Skin
Value
Saturation


4-Skin Classes Graph
Skin
Blackish Skin
Reddish Skin
Non-skin
Value
Saturation


Hue = 354


Raw Data Distribution
Skin
Blackish Skin
Reddish Skin
Light Skin
Non-skin UR
Non-skin LL
Value
Saturation


Skin and Non-Skin classification
Skin
Non-Skin
Value
Saturation


4-Skin Classes Graph
Skin
Blackish Skin
Reddish Skin
Light Skin
Non-skin
Value
Saturation